IBM

704

electronic data-processing machine

manual of operation

MINOR REVISION

This edition, Form 24-6661-2, is a minor revision of the preceding edition but does not obsolete Form 24-6661-1. Principal changes in this edition are:

PAGE	SUBJECT
	Addenda of 24-6661-1 incorporated into text
34	Addition to specification for simultaneous tape writing
6, 7, 10	Reference to new 32,768-word memory
19	Change in specification of ACL operation
22, 26, 29, 92	Change in speed of UFA and SXD operation
27, 29, 32, 36, 91	New ETT operation
32, 36	Physical end of tape
34	New specification for incomplete words on tape
57	Additional information on 716 timing
63	Spacing on 716
66	Additional information on CRT recording unit
84, 87	Octal-decimal integer conversion table extended to five octal digits

Form 24-6661-2

CONTENTS

Magnetic Core Storage — Central Processing Unit — Magnetic Drum, Operator's Console — Power Supply, Printer, Card Reader — Card Punch — Magnetic Tape Units

IBM 704 ELECTRONIC DATA-PROCESSING MACHINES

THE IBM 704 Electronic Data-Processing Machine is a large-scale, high-speed electronic calculator controlled by an internally stored program with instructions of the single address type. This machine is designed for higher speeds and larger capacities required by problems of increasing complexity and size which confront business, industry, government and science. These problems include engineering development, scientific research, production scheduling and control, econometrics, logistics, procurement and supply, and many others.

In order to achieve maximum versatility, every function of the machine is under control of the stored program. This versatility allows the machine to execute instructions at the rate of about 40,000 per second on most problems. Also, the functions of getting data in and out of the calculator are controlled by the stored program, and hence, under the complete control of the operator. The great advantage of this system lies in the fact that a customer may build up a library of programs which will perform his special applications at peak machine efficiency.

To achieve greater computing efficiency, the 704 works internally in the binary number system. The input and output, however, may be accomplished directly on standard IBM cards in the familiar decimal number system by programming which does not interfere with maximum reading, punching, and printing speeds. Or the information on cards may be put on a tape on peripheral equipment and the tape will then be the primary input. Similarly, the results of a computation may be put on a tape and, at some later time, punched on cards or printed by peripheral equipment.

The internal high-speed storage on the 704 is magnetic core storage. When the amount of storage available in magnetic core storage is not large enough, magnetic drums are used to store and supply large blocks of information for ready access at frequent intervals. When the amount of storage needed is in excess of the capacities of both core storage and magnetic drums, then magnetic tapes are used. Also, information may be stored on tapes and the tapes may then be removed from the calculator. In this way, large amounts of information can be filed for future reference in a very compact and convenient form. Magnetic tape is a storage and input-output medium that allows rapid reading and writing and can be reused many times.

The stored programs may be written and introduced into the calculator in many ways. Usually the instructions are key punched on cards in their original form and read into the machine. If the program is to be preserved for future use, it can be punched on cards in the binary number system for compactness or recorded on tape and filed away. To prepare the machine for calculation the appropriate magnetic tapes are inserted in the tape units, cards are placed in the punch hopper, if necessary, and the cards containing the instructions and data of the problem are placed in the hopper of the card reader. By pressing one key the calculator may be made to store the program and data of the problem and start computing. From then on operation of the calculator is fully automatic, with all the components being under the complete control of the program, although it is possible for the operator to interrupt the calculation manually at any time.

All of the real work is done in the central processing unit; that is, all additions, subtractions, multiplications, etc. are done in the special registers of the central processing unit. In addition to standard arithmetic, the 704 has instructions which will perform logical arithmetic for increased flexibility in doing complex problems. Also in the central processing unit are three index registers for automatic counting and effective address modification.

An important feature on the 704 is a complete set of instructions which will perform floating-point arithmetic. This manual includes a complete description of floating-point numbers and the special floating-point instructions (such as floating add, subtract, multiply, divide or halt, and divide or proceed) needed to manipulate data in this form.

STORAGE AND INPUT-OUTPUT UNITS

Access Time

The fundamental machine cycle of the 704 is 12 microseconds. One cycle is the core storage access time, that is, the time required by the central processing unit to transmit or receive a word of information to or from core storage. The time required to transmit information between core storage and any of the input-output units is given in the description of the unit.

Address System

Individual locations (or registers) in magnetic core storage, together with magnetic drums, magnetic tapes, and all input-output units are identified by a system of numerical addresses. By means of a number contained in the *address* part of an instruction, it is possible to refer to the information contained in any register in magnetic core storage or any component of the machine.

Magnetic Core Storage

Information is stored in the primary storage unit by the use of magnetic cores. Each core is a ring of ferromagnetic material. The cores can retain information indefinitely, and recall it in a few millionths of a second. When a wire is inserted through the hollow center of a core, a current passed along the wire sets up a magnetic field around the wire. This magnetizes the core. When the current is removed, the core remains magnetized. If the current is sent along the wire in the opposite direction, the magnetic field set up around the wire is reversed. If the current is again removed, the core will again remain magnetized but its magnetic state will be opposite to that which remained after the first current was removed (Figure 1).

If the first magnetic state can be called positive, the second can be called negative. The positive state can be used to represent a 1; the negative, zero. A group of 36 cores constitutes one register in storage. Magnetic core storage units are available with capacities of either 4,096 or 32,768 core storage registers; or two magnetic core storage units, each with a capacity of 4,096 core storage registers, may be used. Thus, magnetic core storage units are available to give the calculator a capacity of 4,096, 8,192 or 32,768 core storage registers.

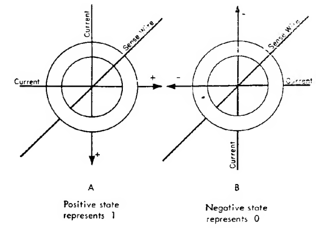


FIGURE 1

The principal advantage of magnetic core storage over other types is the very small time necessary to extract information from any given location and send it to the central processing unit. Also the program has random access to any core storage location. Information is not retained when the power is off.

Magnetic Drum Storage

Additional storage capacity is provided by eight magnetic drums in two drum units. These drums are rotating cylinders surfaced with a material that can be magnetized locally. Binary digits are stored on a drum through the presence or absence of small magnetized areas at cerain locations on the surface of the drum. Each drum has a storage capacity of 2048 words. The location of a word on a drum is identified by a system of addresses analogous to the system used for core storage.

Any part of the information on a drum can be selectively altered at any time. Because access to individual words on a drum is slow in relation to core storage access, it is more efficient to use the drums for storing large blocks of information. After the first word of such a block has been located, the remaining words are transmitted at the rate of 10,000 words per second. Magnetic drums will retain information when the power is turned off.

Magnetic Tapes

For greater internal working storage as well as their input-output function, ten magnetic tape units are available on the 704. Each unit contains one reel of tape which may be 2400 feet long. The tape itself is a plastic, oxide-coated band one-half inch wide. Binary information is recorded on a tape by means of magnetized spots. A block of words recorded consecutively on a tape is called a *record*. The amount of information contained on each tape depends on the lengths of the individual records since there is a certain amount of space between each record to allow for starting and stopping the tape. It is possible to

store as many as 900,000 words on each tape. After the tape is in motion, information can be transmitted at the rate of 2500 words per second.

Flow of Information

The magnetic core storage is always connected to the central processing unit; also, it is the site of the stored program which controls the entire calculator. The auxiliary storage media and the input-output devices, on the other hand, are normally disconnected; they become connected only by the execution of certain stored program instructions. The contents of these units may control the calculator only after being copied into core storage. Thus, information flows between input-output components and magnetic core storage through the central processing unit (Figure 2).

WORDS

In the 704 the word, or basic unit of information, consists of 36 binary digits (36 bits). Words may be stored in 4,096 distinct word locations in each of the smaller magnetic core storage units, in 32,768 distinct word locations in the larger magnetic core storage unit, on magnetic drums (8,192 words per drum unit), on magnetic tapes (33⅓ words per inch of tape), or on punched cards (24 words per card).

A word may be an instruction, a fixed-point number, a floating-point number, or any pattern of 36 bits desired by the programmer for any reason. The 36 positions of a word are shown schematically in Figure 3. S refers to the sign position, *1* refers to bit position 1, *2* refers to bit position 2, and so on.

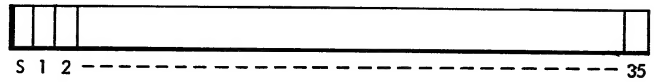


Figure 3

When a word is interpreted as numerical data, the zero position acts as the sign (position S in the diagram) of the word. If the sign position contains a 0, the word is positive; if it contains a 1, the word is negative. When a *logical* operation is performed on a word, the word is interpreted as a 36-bit signless number. As an algebraic (signed) binary number, a word can represent all ten-digit algebraic decimal numbers, and eleven-digit decimal numbers which are less than 34,359,738,368. Three binary digits are exactly equal to one octal digit, and, therefore, a signless word consists of twelve octal digits.

When alphabetic or alphamerical information is being processed with the binary-coded decimal representation shown in Table III, page 35, a word may contain six characters.

Instructions

The two principal classes of instructions are referred to as Types A and B. Figure 4 shows the form

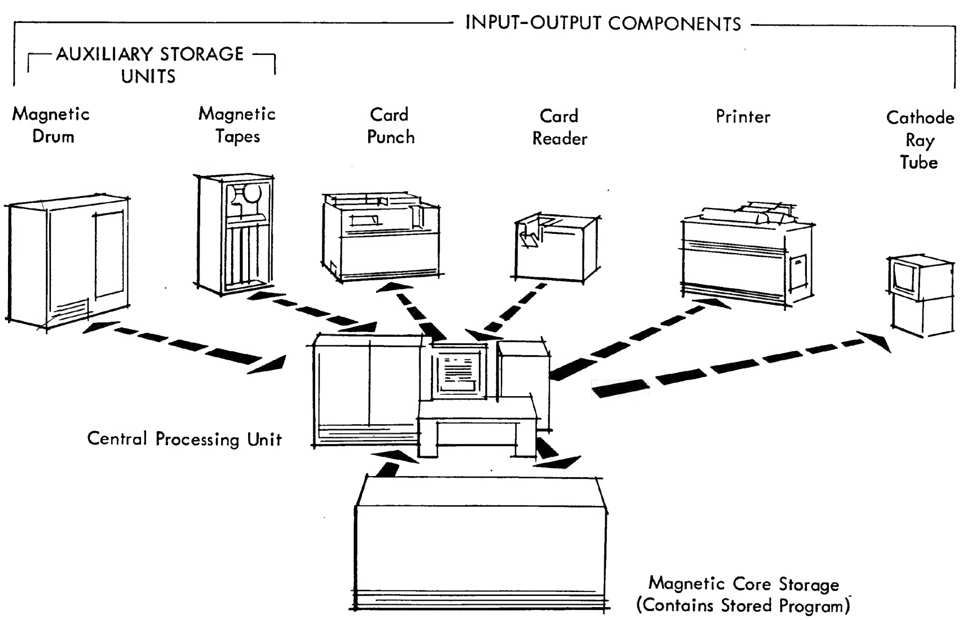


Figure 2

TYPE A INSTRUCTION

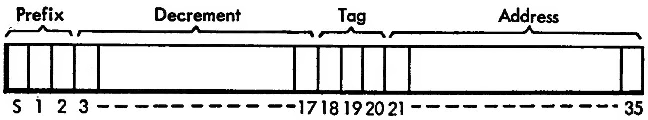


FIGURE 4

of a Type A instruction. Type A instructions use two 15-bit fields (decrement and address) containing numbers in the octal range 00000 to 77777. The prefix contains the operation part while the contents of the tag field select the index register used by the instruction. Positions *1* and *2* of Type A instructions are not both zero.

Bits 21-35 are called the *address* part of an instruction because their principal function is to indicate the storage address of the operand used by the instruction. Bits 3-17 are called the *decrement* part of an instruction because they may represent a number subtracted from the contents of an index register.

Figure 5 shows the form of a Type B instruction. Positions S, *1*, *2*, ...*11*, contain the operation part of Type B instructions, with the exception of the sense-type instructions. These are defined by the code ±0760, and the address part, since they do not refer to a location in storage. Positions 1 and 2 of all Type B instructions are both zero.

Numbers

Numbers are often referred to as data.

Fixed Point. Fixed-point numbers have a sign bit and a magnitude of 35 bits, as illustrated in Figure 6. (Example: The octal fixed-point number + 001367457632 appears in storage as 0 00 000 001 011 110 111 100 101 111 110 011 010.) Theoretically, assume the binary point to be to the right of position 35. However, by proper scale-factoring, the binary point may be placed anywhere in the

TYPE B INSTRUCTION (not sense-type)

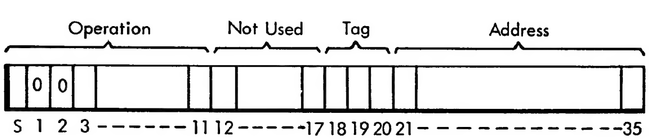


FIGURE 5

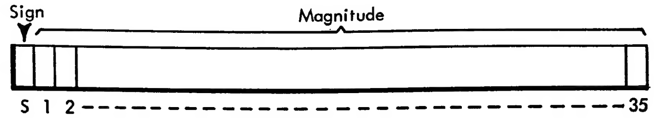


FIGURE 6

number. For example, 0 00 000 . . . 000 010 is equivalent to $1 \times 2^{+1}$.

Floating Point. A floating-point decimal number X may be expressed as a signed proper fraction N times some integral power of 10, or $N \times 10^n$. In the normalized case, the power of ten is chosen so that the decimal point is positioned to the left of the most significant digit of N. Examples:

$\pm$	X	$=$	$\pm$	N	$\times$	$10^{\pm n}$
—	.010	=	—	.10	$\times$	10^{-1}
+	.140	=	+	.14	$\times$	10^0
+	4.600	=	+	.46	$\times$	10^{+1}
—	88.000	=	—	.88	$\times$	10^{+2}

Similarly, a floating-point binary number X may be expressed as a signed proper fraction B times 2^b where b is an integer. In the normalized case the binary point is positioned to the left of the most significant digit of B. Examples:

$\pm$	X	$=$	$\pm$	B	$\times$	$2^{\pm b}$
—	.001	=	—	.100	$\times$	2^{-2}
+	.100	=	+	.100	$\times$	2^0
—	1.100	=	—	.110	$\times$	2^{+1}
+	110.000	=	+	.110	$\times$	2^{+3}

In the 704, a floating-point binary number is stored in a register as shown in Figure 7.

1. The magnitude of B is in bit positions *9-35*. A floating-point binary number having a 1 in position *9* is said to be normalized, (i.e., $1/2 \leqq |B| < 1$).
2. The sign of B is in the S position of the word.
3. Since the sign bit indicates the algebraic sign of the fraction and since signed exponents are desirable, the characteristic, C, of the number, instead of the exponent, is stored in positions *1-8*. The characteristic of the fraction is formed

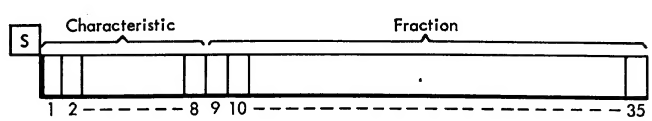


FIGURE 7

by adding $+128$ to the exponent. Thus, the range of the exponent is $-128 \leqq b \leqq 127$, while the range of the characteristic is $0 \leqq C \leqq 255$. (Examples: An exponent of -32 would be represented by a characteristic of $-32 + 128 = +96$. An exponent of $+100$ would be represented by a characteristic of $+100 + 128 = +228$).

CENTRAL PROCESSING UNIT

THE CENTRAL processing unit accomplishes all *arithmetic* and *control* functions. For any given instruction, the time used by the central processing unit to interpret the operation part of the instruction is called the *interpretation* time. The time required to execute an instruction is called the *execution* time. There is some time-sharing between consecutive intructions; that is, while one instruction is being executed, the next instruction is being interpreted, but this rarely concerns the programmer.

Storage Register (SR)

One special register, which will be referred to as the SR, is used for both arithmetic and control functions. Its operation is entirely automatic and will rarely concern the programmer. The SR has a capacity of 36 bits (one word) and serves as a buffer between core storage and the central processing unit. Some of the interpretation of an instruction is performed in the SR. It is also used in the execution of floating-point instructions.

Arithmetic Element

Accumulator (*AC*). The accumulator is a register with a capacity of 37 bits and a sign. See Figure 8.

Nearly every arithmetic operation involves the accumulator. In some operations (for instance, addition, shifting left) it is possible that the contents of the accumulator will overflow positions 1-35. When an overflow occurs, with the exception of overflow caused by the ACL instruction, the AC OVERFLOW

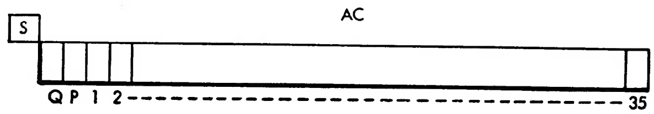


FIGURE 8

indicator is turned on. Certain instructions permit the program to sense the condition of the overflow indicator while the program is being performed. The programmer may preserve some of the overflow information if he wishes. For this purpose, two extra bit positions, or overflow positions, are provided. These are designated the *P* and *Q* positions.

When two numbers having different signs but the same magnitude are added algebraically in the AC, it is important to know if the result is $+0$ or -0, since $+0$ is considered larger than -0. In this case, the sign of the result is identical to the sign of the number in the AC before the addition took place.

Examples: $+6 - (+6) = +0.$
$-6 + (+6) = -0.$

Multiplier-Quotient Register (*MQ*). The MQ is a register with a capacity of 35 bits plus sign. It has five major uses:

1. During the execution of every CPY instruction, the MQ is used as a buffer between core storage and any of the other storage media or input-ouput devices.
2. The multiplier must be placed in the MQ before the execution of a multiplication instruction.
3. After a division instruction is executed, the quotient appears in the MQ (the remainder appears in the AC). In fixed point division, the MQ contains the least significant half of the dividend.
4. After a multiplication instruction is executed, the MQ contains the less significant half of the product. In this connection, the MQ may be regarded as the right-hand extension of the AC; see Figure 9.
5. The least significant 35 bits of the results of FAD, UFA, FSB, and UFS instructions are in the MQ.

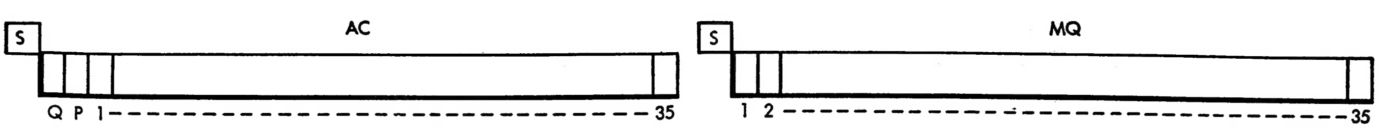


FIGURE 9

Control Element

Instruction Location Counter. This register, with a capacity of 12, 13, or 15 bits (for 4,096, 8,192, or 32,768 words of core storage), determines the location in core storage from which the central processing unit takes its next instruction. After each instruction has been executed, the contents of the instruction location counter are changed. After most instructions, the contents are increased by 1, so that the calculator will go to the next sequential location in storage for its next instruction. However, during the execution of a *skip* type of instruction, the contents may be increased by 1, 2, or 3, and during the execution of a *transfer* type, the contents may be changed to any number in the address range. When the instruction location counter contains the largest possible location in storage (all 1's), then the next sequential instruction is the lowest possible (all 0's).

When operating the 704 and a stop occurs, it is necessary to know the instruction to which the instruction location counter is referring. In all cases except halt and transfer, the instruction location counter contains an address one higher than the address of the last instruction executed. (The last instruction is also the instruction appearing in the instruction register.) In the case of an HTR instruction, the calculator stops with the address of the HTR in the instruction location counter.

Instruction Register. When the central processing unit is ready to accept another instruction, the word in the core storage location specified by the instruction location counter is brought into the SR. In the SR, positions 1 and 2 are tested to determine whether the instruction is Type A or Type B. Depending upon the outcome, the 18 bit positions of the instruction register are filled with the required portions of the instruction word for further interpretation and execution. The instruction register then contains the operation part of the instruction being executed, while the rest of the instruction, i.e., address, tag, and decrement parts, are interpreted in the SR.

With Type A instructions, positions S, 8, 9 of the instruction register contain the contents of positions S, 1, 2 of the instruction. The prefix is the entire operation part of Type A instructions. The remaining positions of the instruction register contain zeros.

With Type B instructions, positions S, 1-9 of the instruction register contain the contents of positions S, 3-11 of the instruction. The remaining positions of the instruction register contain ones with the exception of input-output, shifting and sense instructions. The contents of positions 28-35 of these instructions are placed in positions 10-17 of the instruction register where they are interpreted as part of the operation part of the instruction.

Index Registers. There are three registers, each with a capacity of 12, 13, or 15 bits (for 4,096, 8,192, or 32,768 words of core storage), called index registers A, B, and C. These registers make possible the automatic counting and address modification features of the 704.

With respect to the index registers, the 704 instructions fall into two classes, *non-indexable* and *indexable*.

The non-indexable instructions are the five Type A instructions TIX, TNX, TXH, TXL, TXI and seven of the Type B instructions, namely, TSX, LXA, LXD, SXD, PXD, PAX, and PDX. (Notice that these are the only instructions with an X in the operation code.) Instructions of this class are used to test and manipulate the contents of the index register specified in their tag field.

All other instructions are indexable instructions in their normal form. They are recognizable by the fact that position 8 or 9, or both, contain a zero. If an indexable instruction specifies an index register (that is, one of the three bits in its tag field is a 1), it is executed as if its address field had contained its stated address *minus* the contents of the specified index register. Suppose, for example, that index register B contains 0117_8 and that the instruction CLA B 2117_8, contained in location 1000, is executed. After the execution, the accumulator will contain the contents of core storage location 2000_8. However, the contents of location 1000 are still CLA B 2117_8. This is called *effective* address modification; that is, the address of the instruction is modified in the control unit for execution purposes but is unaltered in storage.

If an instruction specifies no index register (all three bits in its tag field are zeros), it is executed as if the index registers did not exist. Thus CLA 2117_8 will place the contents of core storage location 2117_8 in the accumulator, regardless of the contents of the index registers.

Note that in the case of the fourteen sense-type instructions, effective address modification may actually cause operation modification, because the last

eight bits of these instructions are part of their operations. For example, if index register A contains 0001_8, then SSP is executed as SSP, but SSP A is executed as CHS. (See instructions for octal code of SSP and CHS.)

An instruction may refer to more than one index register by placing multiple 1's in the tag field, such as 011 (when programming, this number must be written in octal form). An instruction (except a fixed instruction) with this tag is executed as if there were a single index register, equivalent to index registers A and B connected in logical OR fashion. For example, if index registers A and B contain 3204_8 and 3631_8, respectively, the instruction CLA 3 6521_8 is executed with an effective address $6521_8 - 3635_8 = 2664_8$. Similarly, the instruction LXD 3 1641_8 causes the contents of both index registers A and B to be replaced by the contents of the decrement part of core storage location 1641_8.

The tag field specifies one or more of the three index registers or no index register as follows:

Tag Field Binary	Tag Field Octal	Index Register(s) Specified
000	0	None
001	1	A
010	2	B
100	4	C
011	3	A OR B
101	5	A OR C
110	6	B OR C
111	7	A OR B OR C

A non-indexable instruction with a zero tag is executed as if there were an imaginary index register always containing zeros. For example, PXD with a tag of zero clears the entire AC; SXD with a tag of zero clears the decrement field of the storage location to which it refers.

Special Indicators and Sense Devices

All special indicators are either on or off. The condition of a particular indicator is tested by means of a test instruction peculiar to that indicator. If an indicator is on when tested, it is turned off by the test. All indicators have a corresponding light on the console for visual checking. The sense lights appearing on the console function in a similar manner. All of these indicators are turned off by manually pressing either the reset or the clear key on the console.

Accumulator Overflow Indicator. This indicator is turned on whenever a 1 passes into or through position P from position 1 of the AC as a result of the execution of an instruction (for example, a carry resulting from algebraic addition). There is no indicator between positions *P* and *Q*, however. Either of the instructions TOV or TNO tests the condition of the AC overflow indicator. The subsequent program is selected according to the outcome of the test.

NOTE: A carry resulting from the instruction ACL does not turn on this indicator.

Multiplier-Quotient Overflow Indicator. This indicator is turned on when a floating-point operation attempts to produce a result with a characteristic C outside the range 000 — 255, inclusive. At any later time, it may be tested and turned off by the TQO instruction.

Divide-Check Indicator. In fixed-point division, this indicator is turned on if the magnitude of the number in the AC (the dividend) is greater than or equal to the magnitude of the number in storage (the divisor). In floating-point division, a divide-check can occur only when the divisor is unnormalized or zero. The divide-check indicator may be tested and turned off by the DCT instruction. It is also turned off by pressing the reset or clear key on the console if the calculator has stopped on a divide check.

Tape Check Indicator. When the calculator or peripheral equipment writes on magnetic tapes, both lateral and longitudinal check (redundancy) bits are automatically written with each unit record. When the tapes are read, the redundancy information is automatically recalculated and compared with the redundancy information stored on the tape. A discrepancy turns on the tape-check indicator. The tape check indicator may be tested and turned off by the RTT instruction.

Trapping Mode Indicator. The 704 can be operated in either of two modes, normal or trapping. Entrance into the trapping mode is made by executing the instruction ETM. Exit from the trapping mode is made by executing the instruction LTM or by manually pressing either the clear or reset key on the console. When the machine is in the trapping mode, the location of each transfer instruction met replaces the address part of location 0000. Unconditional

transfers, and conditional transfers for which the condition is met, are not executed; instead, control is transferred to location 0001_8. One transfer instruction only, TTR, is immune to the trapping mode. The major use of the trapping mode is in program testing, where it permits observation of the flow of control.

Sense Switches. On the console are six switches, which the operator can set either ON or OFF. The condition of any switch may be tested by the PSE instruction with the appropriate address, and the subsequent program selected according to the outcome of the test.

Sense Lights. Also on the console are four sense lights which are turned on by the PSE instruction with the appropriate address. (All four lights are turned off by PSE 140_8). Any sense light can be tested and turned off by the MSE instruction with the appropriate address. The subsequent program can be selected according to the outcome of the test.

INSTRUCTION TYPES

INSTRUCTIONS are divided into two types, A and B, such that only Type A instructions use the decrement field for storing constants within the instruction.

Type A Instructions

There are five Type A instructions in all, namely, TIX, TNX, TXH, TXL, TXI. In the Type A instructions, the 36 bits of the word are divided into four fields—prefix, decrement, tag, and address—as shown in Figure 10.

The prefix is the operation part of the Type A instruction. Type A instructions are distinguished from all others by the fact that bits 1 and 2 are not both zero. The tag denotes the index register(s) to be used in connection with the instruction. See "Central Processing Unit—Index Registers." The decrement field contains a number to be used in connection with the index register specified by the tag. See explanations of individual Type A instructions in "Instructions." The address field refers to a location in core storage. Example:

	PREFIX	DECREMENT	TAG	ADDRESS
Actual binary word in calculator	110	000 100 011 111 101	010	000 110 101 111 001
Equivalent in octal	-2	0 4 3 7 5	2	0 6 5 7 1
Form used in coding (decimal)	TNX	02301	B	03449

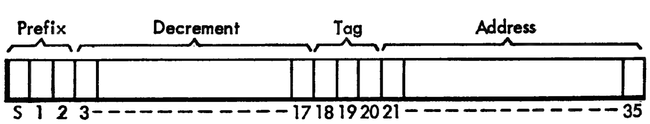


FIGURE 10

Type B Instructions

In Type B instructions, the 36 bits of the word are divided as shown in Figure 11. All Type B instructions have zeros in bits 1 and 2. The sign position and the decrement field contain the operation code. The tag and address fields have the same meaning as in Type A instructions. Example:

	OPERATION	NOT USED	TAG	ADDRESS
Binary	000 110 000 010	xxx xxx	100	000 101 001 010 100
Octal	+0 6 0 2		4	0 5 1 2 4
Coding (decimal)	SLW		C	2 6 4 4

With certain instructions (the shift instructions LLS, LRS, ALS, ARS, LGL, RQL, and the instructions RDS, WRS, BST, WEF, and REW, which are concerned with the input-output units), positions 21-27 of the address field are not interpreted, thus reducing the address field to the last eight bits. For this reason, the address of any of these instructions is interpreted modulo 400_8 = modulo 256_{10}. Example:

	OPERATION	NOT USED	TAG	ADDRESS
Binary	000 111 110 111	xxx xxx	000	xxx xxx x 01 101 100
Octal	+0 7 6 7		0	1 5 4
Coding (decimal)	ALS			1 0 8

The instructions PSE, MSE, CLM, LBT, PBT, CHS, SSP, SSM, COM, ETM, LTM, RND, DCT, and RTT form a special class of Type B instructions, referred to as

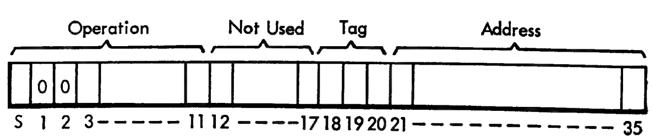


FIGURE 11

sense-type instructions. The contents of position S, 1-11 of these instructions are always ±0760, which together with the contents of positions 28-35 form the operation part. The contents of positions 21-27 are not interpreted when executing these instructions. Examples:

	OPERATION	NOT USED	TAG	ADDRESS
Binary	100 111 110 000	xxx xxx	000	xxx xxx x 01 100 011
Octal	—0 7 6 0		0	1 4 3
Coding (decimal)	MSE			0 9 9

	OPERATION	NOT USED	TAG	ADDRESS
Binary	100 111 110 000	xxx xxx	000	xxx xxx x 00 000 011
Octal	—0 7 6 0		0	0 0 3
Coding (decimal)	SSM			0 0 3

MANUAL OPERATION

FIGURE 12 shows the operator's console which includes indicating lights and operating keys.

Panel Lights

Internal Register Display. The contents of the internal registers (instruction location counter, instruction register, SR, AC, MQ) are displayed directly on the 704 operator's console by neon glow tubes, one for each bit position (a light on represents a 1; a light off represents a 0).

Index Register Display. A row of 15 lights can display the contents of any one of three index registers, depending on which one of a set of three panel keys is pressed. See "Panel Keys."

Trap Indicator Light. The trap indicator light goes on when the calculator is operating in the trapping mode.

Sense Lights. There are four sense lights which may be turned on and off by the program. They are explained under plus sense and minus sense instructions.

Program Stop Light. The program stop light is turned on when the calculator executes a halt instruction and stops.

Accumulator Overflow Light. The accumulator overflow light is on or off when the accumulator overflow indicator is on or off.

MQ Overflow Light. The MQ overflow light is on or off when the MQ overflow indicator is on or off.

Divide-Check Light. The divide-check light turns on or off when the divide-check indicator is turned on or off.

Read-Write Select Light. The read-write select light goes on when one of the input-output units has been selected for reading or writing. The light goes off when the input-output unit is disconnected and no other input-output unit is selected.

Read-Write Check Light. The read-write check light goes on and the calculator halts when a copy and skip instruction is given at an inadmissible time. See "Instructions."

Tape-Check Light. The tape-check light is on or off when the tape check indicator is on or off.

Ready and Power Lights. The ready and power lights are on when the calculator is ready to begin operating.

Automatic Light. The automatic light is on when the calculator is executing instructions in the automatic mode of operation (as distinct from the manual mode).

Panel Keys and Switches

Automatic-Manual Switch. Pressing the automatic-manual switch stops the calculator after it has completed the execution of the instruction then being processed, unless an input-output unit is connected to the logical unit. In this case, the calculator stops after the input-ouput unit in use has been disconnected. The automatic light goes out and all of the switches and the following keys become effective: enter MQ, enter instruction, display storage, display effective address, display A, display B, display C, multiple step, and single step. The clear key becomes ineffective.

Single Step and Multiple Step Keys. These keys enable the operator, when the calculator is on MANUAL, to proceed with his program either one step at a time or at a very low rate of speed. If an instruction is executed to cause an input-output unit to be connected to the calculator, the calculator operates in the automatic mode until the input-output unit is disconnected. When this occurs, the calculator returns to the manual mode.

Sense Switches. Six sense switches give the operator manual control over the program while it is being executed by the calculator at high speed. At various

FIGURE 12

points in the program, giving sense instructions (explained under "Instructions") with the addresses of the sense switches causes the calculator to follow one of two courses, depending on which sense switches are depressed. The sense switches are also effective while the calculator is on MANUAL.

Panel Input Switches. There are 36 panel input switches, enabling the operator to insert a word of information into various registers of the calculator while it is on MANUAL. When a panel input switch is down, it represents a 1; when up, it represents a 0.

Index Display Keys. The three index display keys let the operator display the contents of any of the index registers, while the calculator is on MANUAL, by pressing the key marked with the letter corresponding to the index register in question. For example, to display the contents of index register A, the operator presses the key marked DISPLAY A; the contents of index register A then appears in the index lights. The index registers are automatically displayed until the calculator is returned to automatic operation.

More than one index register can be manually displayed in sequence by pressing the Display A, Display B, and Display C keys, in that order. No return to the automatic mode is necessary.

Load Keys. The load keys let the operator initiate the loading of a self-loading program stored on

binary cards, a drum, or on a tape. If a self-loading program is stored on the tape whose logical identification is 145, pressing the load-tape key causes the calculator to perform the following sequence of four instructions after resetting the read-write check light.

Read Select	145 (decimal)
Copy and skip	0
Copy and skip	1
Transfer	0

See "Instructions" for an explanation of these operations. This sequence of instructions starts the loading of a self-loading program stored on tape 145.

Pressing the load-card key causes the same sequence of instructions to be executed, except that the address in the first instruction is 209, selecting the card reader. A similar situation holds for the drum, with an address of 193.

When the loading is initiated, it is essential that the particular input unit from which information is to be loaded into storage be in a ready status (indicated by a light on the input unit).

Reset Key. Pressing the reset key resets all registers and indicators in the logical section of the machine. That is, the SR, AC, MQ, instruction location counter, instruction register, and index registers are set to zero and all indicators are turned off. The panel lights are all turned off with the exception of those marked POWER and READY. Core storage is *not* affected by the reset key.

Clear Key. If the calculator is on AUTOMATIC, pressing the clear key resets all the registers in core storage. The entire logical section is reset, just as if the reset key had been depressed also. The clear key is ineffective when the calculator is on MANUAL.

Start Key. Pressing the start key continues calculation at high speed if the calculator has halted at a program stop, a read-write check, or if it has been returned to automatic operation after having been on MANUAL. This turns off the read-write check light and starts calculation, starting with the operation then in the instruction register.

Enter MQ Key. If the operator manually keys a given word of information into the panel input switches and if the enter MQ key is pressed while the calculator is on MANUAL, then the keyed-in word replaces the contents of the MQ. The contents of the SR are destroyed by this operation.

Enter Instruction Key. If the operator presses the enter instruction key under the same conditions as above, then the operation part of the keyed-in word goes into the instruction register, the full word is placed in the SR, and the instruction is executed.

Display Storage Key. If, while the calculator is on MANUAL, the operator keys the location into the address part of the panel input switches, and presses the display storage key, the contents of the keyed-in location go into the SR where they may be read from the SR lights.

Display Effective Address Key. Assume the calculator is on MANUAL and the display effective address key is pressed. The difference between the contents of the address field of the instruction in the SR and those of the index register tagged in that instruction (if one is tagged) will appear in the address field of the SR where it may be read from the SR lights. If any index registers have been displayed prior to displaying effective address, put the automatic-manual switch on AUTOMATIC and then back on MANUAL before pressing the display effective address key.

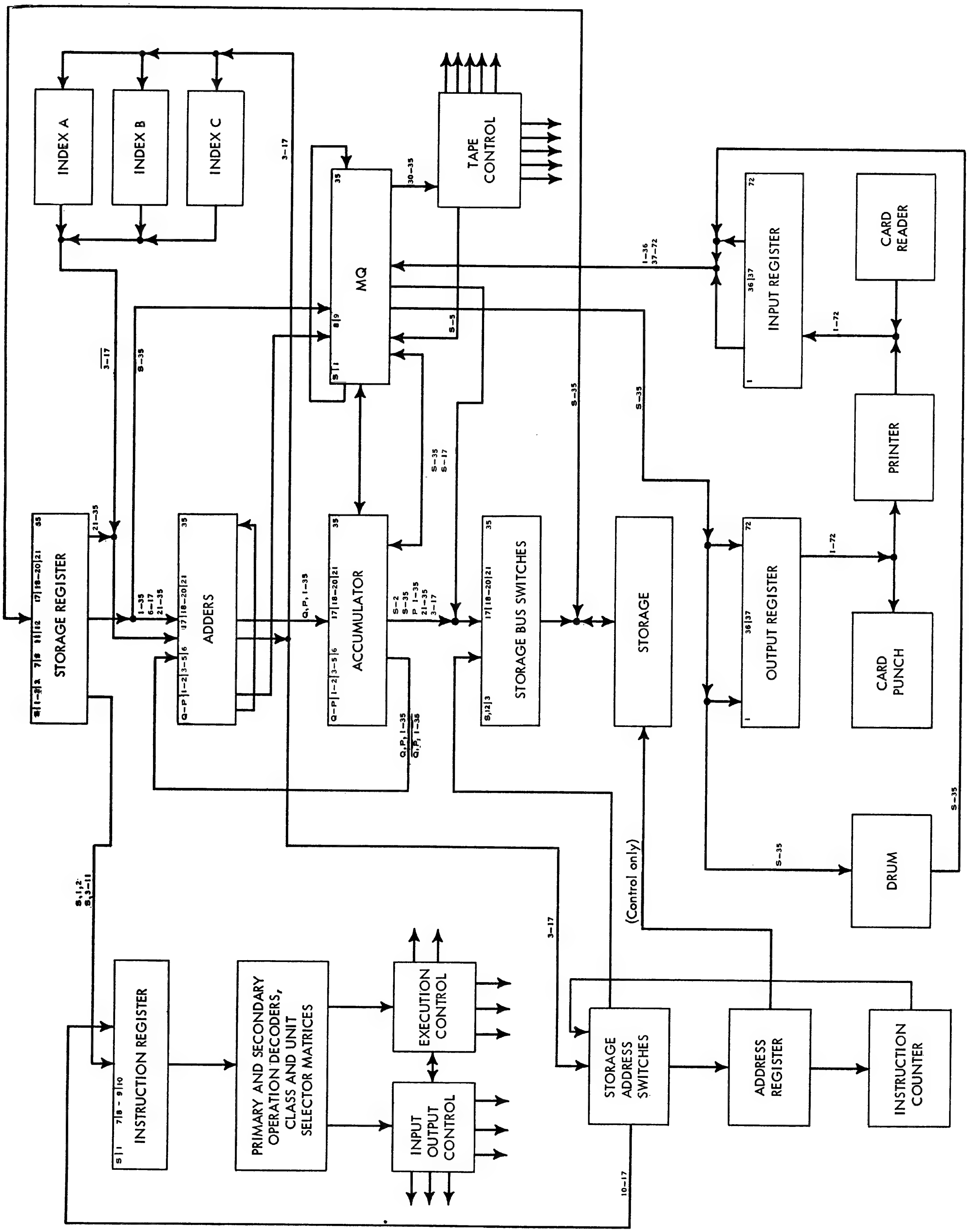
The circuitry for displaying the effective address does not distinguish between instruction types; hence even the address of type A instructions will appear as an "effective address."

Power-on, Normal-off, DC-on and DC-off Keys. These keys assist the maintenance engineers servicing the calculator and have no programming significance.

CENTRAL PROCESSING DIAGRAM

THE BLOCK diagram (Figure 13) describes the flow of information within the central processing unit. Many registers in this diagram are not mentioned in the preceding or succeeding sections because they do not concern the programmer directly. They are included here to help those who wish to better understand the operation of the 704.

The connecting lines betwen registers indicate the flow of information between these registers. The numbers associated with the lines indicate the bit positions of the registers from which the information is obtained. The bar over the numbers indicates that the complement of the number in the initial register is transmitted to the receiving register.



INDEX A
INDEX B
INDEX C
TAPE CONTROL
MQ
STORAGE REGISTER
ADDERS
ACCUMULATOR
STORAGE BUS SWITCHES
STORAGE
INPUT REGISTER
CARD READER
PRINTER
OUTPUT REGISTER
CARD PUNCH
DRUM
INSTRUCTION REGISTER
PRIMARY AND SECONDARY OPERATION DECODERS, CLASS AND UNIT SELECTOR MATRICES
EXECUTION CONTROL
INPUT OUTPUT CONTROL
STORAGE ADDRESS SWITCHES
ADDRESS REGISTER
INSTRUCTION COUNTER
(Control only)
3–17
S–35
30–35
S–5
S–35
S–17
1–36
37–72
1–72
S–35
10–17

FIGURE 13

The contents of positions S, 1-5 of the instruction register are called the *primary operation* part while the contents of positions 6-9 of the instruction register are called the *secondary operation* part. The class and unit selector matrices are used with input-output instructions. The class refers to drum, tape, printer, and so on. The unit selector matrix selects which drum or tape is to be used.

INSTRUCTIONS

THIS SECTION states in the heading for each instruction the title, the number of fundamental cycles required for the execution of the instruction, the three-letter alphabetic code for the instruction, and the numerical code for the instruction. The number of fundamental cycles required may be modified if followed by a Roman numeral, I, II, III, or IV; see "Timing." If the instruction has an address part associated with it in normal operation, this section uses the letter Y to denote that address part. Y may be an address in storage, the length of a shift, or the address of an input-output unit. The numerical code is given in the octal number system because this can be visually converted to binary for reference to the various bit patterns which the calculator interprets. The sign and four octal digits correspond to positions S, 1-11 of all instructions. For the sense instructions with a fixed address part (e.g. the operation round), the three octal digits corresponding to positions 28-35 of the address part are separated from the operation part by three dots (e.g., +0760. . .010).

Note the following definitions:

1. C(Y) denotes the contents of location Y, when Y refers to some location in storage. C(SR) denotes the contents of the storage register. Similarly, C(AC) denotes the contents of the accumulator. In addition, subscripts refer to the individual bit positions of a register; i.e., $C(MQ)_{S,1\text{-}17}$ is read "the contents of positions S, 1, 2, . . ., 17 of the MQ." When subscripts are not used with this notation, the entire register is implied; i.e., C(AC) implies positions S, Q, P, 1-35 inclusive.
2. Instructions which have a decrement part are indicated in the explanations. All instructions are indexable unless the explanation specifically states that the instruction is non-indexable. All non-indexable instructions have an X in their alphabetic code.
3. When a register or part of a register is cleared, the cleared part is reset to 0's just as storage is reset to 0's when the clear key on the console is depressed.
4. The negative of a number is the number with its sign reversed.
5. The magnitude of a number is the number with its sign made positive (a 0 in position S corresponds to a positive sign).
6. The complement of a binary number is defined as the number derived by replacing all 1's with 0's and all 0's with 1's. Therefore, all binary bits are inverted to produce the complement of a number.
7. The 2's complement of a binary number is 1 plus the complement of the number.
8. When the words "store" or "load" are used in the title of an instruction, magnetic core storage is always one of the agents; for example, "store address."
9. When the word "place" is used in the title of an instruction, the AC is always one of the agents; for example, "place address in index."
10. In the three-letter operation code:
    a. The letter Q designates the MQ register.
    b. The letter X designates an index register.
    c. The first letter of all transfer instructions is a T.
    d. The last letter of all test instructions is a T.

Fixed-Point Arithmetic Operations

The following instructions refer to arithmetic operations using fixed-point data.

Clear and Add
2 CLA Y +0500

The C(Y) replace the $C(AC)_{S,1\text{-}35}$. Positions Q and P of the AC are cleared. The C(Y) are unchanged.

Add
2 ADD Y +0400

This instruction algebraically adds the C(Y) to the C(AC) and replaces the C(AC) with this sum. The C(Y) are unchanged. AC overflow is possible.

Add Magnitude
2 ADM Y +0401

This instruction algebraically adds the magnitude of the C(Y) to the C(AC) and replaces the C(AC) with this sum. The C(Y) are unchanged. AC overflow is possible.

Clear and Subtract
2 CLS Y +0502

The negative of the C(Y) replaces the $C(AC)_{S,1\text{-}35}$. Positions Q and P of the AC are cleared. The C(Y) are unchanged.

Subtract
2 SUB Y +0402

This instruction algebraically subtracts the C(Y) from the C(AC) and replaces the C(AC) with this difference. The C(Y) are unchanged. AC overflow is possible.

Subtract Magnitude
2 SBM Y —0400

This instruction algebraically subtracts the magnitude of the C(Y) from the C(AC) and replaces the C(AC) with this difference. The C(Y) are unchanged. AC overflow is possible.

Multiply
20 MPY Y +0200

This instruction multiplies the C(Y) by the C(MQ). The 35 most significant bits of the 70-bit product replace the $C(AC)_{1\text{-}35}$ and the 35 least significant bits replace the $C(MQ)_{1\text{-}35}$. The Q and P bits are cleared. The sign of the AC is the algebraic sign of the product. The sign of the MQ agrees with the sign of the AC.

Placing of the binary point in the factors is completely arbitrary. A simple familiar rule to remember with regard to placing the binary point in the resulting product follows.

RULE: Add the number of binary bits to the right of the binary point in the first factor to the number of binary bits to the right of the binary point in the second factor. This sum is the number of bits appearing to the right of the binary point in the product.

Multiply and Round
20 MPR Y —0200

This instruction executes a multiply followed by a round. (The latter operation is defined below.) AC overflow is not possible.

Round
2 RND +0760 . . . 010

If position 1 of the MQ contains a 1, the magnitude of the C(AC) is increased by a 1 in position 35. If position 1 of the MQ contains a zero, the C(AC) remain unchanged. In either case, the C(MQ) are unchanged. AC overflow is possible.

Divide or Halt
20 DVH Y +0220

This instruction treats the $C(AC)_{S,Q,P,1\text{-}35}$ and the $C(MQ)_{1\text{-}35}$ as a 72-bit dividend plus sign, and the C(Y) as the divisor. If $|C(Y)| > |C(AC)|$, division takes place, a 35-bit quotient plus sign replaces the C(MQ) and the remainder replaces the $C(AC)_{S,1\text{-}35}$. The sign of the remainder always agrees with the sign of the dividend.

If $|C(Y)| \leqq |C(AC)|$, division does not take place and the calculator stops with the divide-check indicator and light on. Consequently, if position Q or P of the AC contains a 1, division does not take place since $|C(Y)| < |C(AC)|$. The dividend remains unchanged in the AC.

The binary point is placed as follows:

"Standard" Case: Assume that the binary point of the dividend is located between position 35 of the AC and the first position of the MQ. Also assume that the divisor has its binary point to the right of position 35. Then the quotient will have the binary point located to the left of position 1 of the MQ. The remainder has its binary point located between positions P and 1 of the AC.

Variations of the standard case are:

Rule 1: A change in the binary point in the dividend results in a change equal in magnitude and in the same direction in the binary points of both the quotient and the remainder.

Rule 2: A change in the binary point of the divisor results in a corresponding change in the opposite direction of the binary point in the quotient. The binary point of the remainder is unchanged.

Divide or Proceed
20 DVP Y +0221

This instruction executes a division (as defined above) if $|C(Y)| > |C(AC)|$. If $|C(Y)| \leqq |C(AC)|$, division does not take place, the divide-check indicator and light are turned on, and the calculator pro-

ceeds to the next instruction. The dividend remains unchanged in the AC.

Load MQ
2 LDQ Y +0560

The C(Y) replace the C(MQ). The C(Y) are unchanged.

Store MQ
2 STQ Y —0600

The C(MQ) replace the C(Y). The C(MQ) are unchanged.

Store Left-Half MQ
2 SLQ Y —0620

The $\text{C(MQ)}_{S,1\text{-}17}$ replace the $\text{C(Y)}_{S,1\text{-}17}$. The $\text{C(Y)}_{18\text{-}35}$ and the C(MQ) are unchanged.

Store
2 STO Y +0601

The $\text{C(AC)}_{S,1\text{-}35}$ replace the $\text{C(Y)}_{S,1\text{-}35}$. The C(AC) are unchanged.

Store Prefix
2 STP Y +0630

The $\text{C(AC)}_{P,1,2}$ replace the $\text{C(Y)}_{S,1,2}$. The $\text{C(Y)}_{3\text{-}35}$ and the C(AC) are unchanged.

Store Decrement
2 STD Y +0622

The $\text{C(AC)}_{3\text{-}17}$ replace the $\text{C(Y)}_{3\text{-}17}$. The $\text{C(Y)}_{S,1,2,18\text{-}35}$ and the C(AC) are unchanged.

Store Address
2 STA Y +0621

The $\text{C(AC)}_{21\text{-}35}$ replace the $\text{C(Y)}_{21\text{-}35}$. The $\text{C(Y)}_{S,1\text{-}20}$ and the C(AC) are unchanged.

Clear Magnitude
2 CLM +0760 . . . 000

The $\text{C(AC)}_{Q,P,1\text{-}35}$ are cleared. The AC sign is unchanged.

Change Sign
2 CHS +0760 . . . 002

If the AC sign bit is negative, it is made positive, and vice versa.

Set Sign Plus
2 SSP +0760 . . . 003

A positive sign replaces the C(AC)_{S}.

Set Sign Minus
2 SSM —0760 . . . 003

A negative sign replaces the C(AC)_{S}.

Logical Operations

Clear and Add Logical Word
2 CAL Y —0500

This instruction replaces the $\text{C(AC)}_{P,1\text{-}35}$ with the C(Y). Thus the sign of the C(Y) appears in position P of the AC, and the S and Q bits are cleared. The C(Y) are unchanged.

Add and Carry Logical Word
2 ACL Y +0361

This instruction adds the $\text{C(Y)}_{S,1\text{-}35}$ to the $\text{C(AC)}_{P,1\text{-}35}$, respectively, and replaces the $\text{C(AC)}_{P,1\text{-}35}$ with this sum (position S of register Y is treated as a numerical bit, and the sign of the AC is ignored). A carry out of the P bit adds into position 35 of the AC, but does not add into Q. Q is not changed. The C(Y) are unchanged. No overflow is possible. See Figure 14.

Store Logical Word
2 SLW Y +0602

The $\text{C(AC)}_{P,1\text{-}35}$ replace the $\text{C(Y)}_{S,1\text{-}35}$. The C(AC) are unchanged.

AND to Accumulator
3 ANA Y —0320

Each bit of the $\text{C(AC)}_{P,1\text{-}35}$ is matched with the corresponding bit of the $\text{C(Y)}_{S,1\text{-}35}$, the C(AC)_{P} being

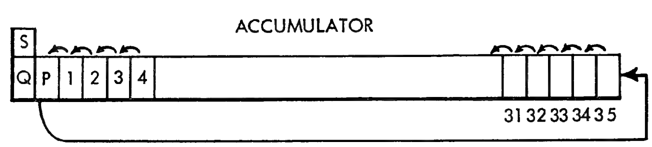


FIGURE 14

matched with the $C(Y)_S$. When the corresponding bit of both the AC and location Y is a one, a one replaces the contents of that position in the AC. When the corresponding bit of either the AC or location Y is a zero, a zero replaces the contents of that position in the AC. The $C(AC)_{S,Q}$ are cleared. The $C(Y)$ are unchanged.

AND to Storage
4 ANS Y +0320

Each bit of the $C(AC)_{P,1\text{-}35}$ is matched with the corresponding bit of the $C(Y)_{S,1\text{-}35}$. The $C(AC)_P$ being matched with the $C(Y)_S$. When the corresponding bit of both the AC and location Y is a one, a one replaces the contents of that position in location Y. When the corresponding bit of either the AC or location Y is a zero, a zero replaces the contents of that position in location Y. The $C(AC)$ are unchanged.

OR to Accumulator
2 ORA Y —0501

Each bit of the $C(AC)_{P,1\text{-}35}$ is matched with the corresponding bit of the $C(Y)_{S,1\text{-}35}$, the $C(AC)_P$ being matched with the $C(Y)_S$. When the corresponding bit of either the AC or location Y is a one, a one replaces the contents of that position in the AC. When the corresponding bit of both the AC and location Y is a zero, a zero replaces the contents of that position in the AC. The $C(Y)$ and the $C(AC)_{S,Q}$ are unchanged.

OR to Storage
2 ORS Y —0602

Each bit of the $C(AC)_{P,1\text{-}35}$ is matched with the corresponding bit of the $C(Y)_{S,1\text{-}35}$, the $C(AC)_P$ being matched with the $C(Y)_S$. When the corresponding bit of either the AC or location Y is a one, a one replaces the contents of that position in location Y. When the corresponding bit of both the AC and location Y is a zero, a zero replaces the contents of that position in location Y. The $C(AC)$ are unchanged.

Complement Magnitude
2 COM +0760 . . . 006

All ones are replaced by zeros and all zeros are replaced by ones in the $C(AC)_{Q,P,1\text{-}35}$. The AC sign is unchanged.

Shifting Operations

Shift instructions are used to move the bits in a word to the right or left of their original positions in the AC or MQ register or both. With the exception of the RQL instruction, zeros are automatically introduced in the vacated positions of a register. Thus, a shift larger than the bit capacities of the registers involved in the shifting will have no significance after the capacities of the registers are exceeded. When an instruction is interpreted as a shift instruction, the extent of the shift is determined by the least significant eight bits of the address of the instruction. Since the maximum possible shift is 255, a number larger than 255 in the address part of a shift instruction is interpreted modulo 256.

Example 1: 583 modulo 256 = 71
 [because 583 = 2(256) + 71]
Example 2: 256 modulo 256 = 0
Example 3: 15 modulo 256 = 15

Shifting a number in a register is equivalent to multiplying that number by a power of 2 (as long as none of the significant bits are lost).

Example 1: Shifting a binary number three places to the left is equivalent to multiplying it by 2^3.

Example 2: Shifting a binary number five places to the right is equivalent to multiplying it by 2^{-5}.

Accumulator Left Shift
2-I ALS Y +0767

The $C(AC)_{Q,P,1\text{-}35}$ are shifted left Y modulo 256 places. If a non-zero bit is shifted into or through position P, the AC overflow indicator and light are turned on. Bits shifted past position Q are lost. Positions made vacant are filled in with zeros.

Accumulator Right Shift
2-I ARS Y +0771

The $C(AC)_{Q,P,1\text{-}35}$ are shifted right Y modulo 256 places. Bits shifted past position 35 of the AC are lost. Positions made vacant are filled in with zeros.

Long Left Shift
2-I LLS Y +0763

The $C(AC)_{Q,P,1\text{-}35}$ and the $C(MQ)_{1\text{-}35}$ are shifted left Y modulo 256 places. Bits enter position 35 of the AC from position 1 of the MQ. If a non-zero bit is

shifted into or through position P, the AC overflow indicator and light are turned on. Bits shifted past position Q are lost. Positions made vacant are filled in with zeros. The sign of the AC is replaced by the same sign as that of the MQ. See Figure 15.

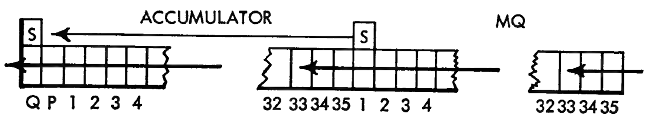


FIGURE 15

Long Right Shift
2-I LRS Y +0765

The $\text{c(AC)}_{Q,P,1\text{-}35}$ and the $\text{c(MQ)}_{1\text{-}35}$ are shifted right Y modulo 256 places. Bits enter position 1 of the MQ from position 35 of the AC. Bits shifted past position 35 of the MQ are lost. Positions made vacant are filled in with zeros. The sign of the MQ is replaced by the same sign as that of the AC.

Logical Left
2-I LGL Y —0763

The $\text{c(AC)}_{Q,P,1\text{-}35}$ and the $\text{c(MQ)}_{S,1\text{-}35}$ are shifted left Y modulo 256 places. Bits enter position S of the MQ from position 1 of the MQ, and enter position 35 of the AC from position S of the MQ. If a nonzero bit is shifted into or through position P of the AC, the AC overflow indicator and light are turned on. Bits shifted past position Q are lost. Positions made vacant are filled in with zeros. The sign of the AC is unchanged. See Figure 16.

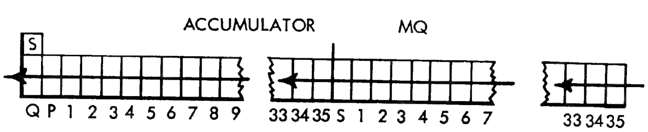


FIGURE 16

Rotate MQ Left
2-I RQL Y —0773

The $\text{c(MQ)}_{S,1\text{-}35}$ are rotated left Y modulo 256 places. The bits rotate from position 1 to position S of the MQ, and from position S to position 35 of the MQ. See Figure 17.

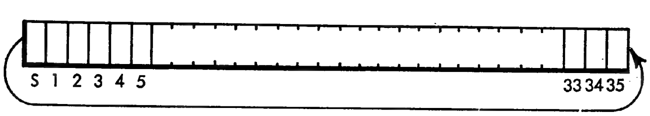


FIGURE 17

Floating-Point Arithmetic Operations

Floating Add
7-II FAD Y +0300

The c(Y) are algebraically added to the c(AC), and this sum replaces the c(AC) and the c(MQ). The c(Y) are unchanged. Floating-point addition takes place in the following way:

1. The MQ is cleared.

2. The c(Y) are placed in the SR.

3. If the characteristic in the SR is less than the characteristic in the AC, the c(SR) and the c(AC) interchange automatically because the number with the smaller characteristic must appear in the AC before addition can take place. (Positions Q and P of the AC are considered as part of the characteristic. Consequently, a 1 in either of these positions makes the characteristic in the AC larger than that in the SR, but the 1's would be lost during the interchange and an incorrect answer will result.)

4. The MQ is given the same sign as the AC.

5. The fraction in the AC is shifted right the number of positions equal to the magnitude of the difference in the characteristics. Bits shifted out of the AC enter position 9 of the MQ. Bits shifted out of position 35 of the MQ are lost.

6. The characteristic in the SR replaces the $\text{c(AC)}_{1\text{-}8}$.

7. The fraction in the SR is algebraically added to the fraction in the AC and this sum replaces the $\text{c(AC)}_{S,9\text{-}35}$.

8. If the magnitude of the sum is greater than or equal to 1, there is a carry from position 9 to position 8 of the AC (thus increasing the characteristic by 1).* In this event, the $\text{c(AC)}_{9\text{-}35}$ and the $\text{c(MQ)}_{9\text{-}35}$ are shifted right one position and 1 is inserted in position 9 of the AC.

9a. If the resulting fraction in the AC is zero, the AC is cleared, yielding a normal zero. The sign of the AC is the sign of the number that has the smaller characteristic. If both characteristics are equal, then the sign of the AC is the sign of the number in the AC.

9b. If the magnitude of the resulting fraction in the AC is not in normal form (i.e. less than ½ but not zero), and the signs of the MQ and AC are the same, the $\text{c(AC)}_{9\text{-}35}$ and the $\text{c(MQ)}_{9\text{-}35}$ are shifted left until a 1 is in position 9 of the AC. Bits enter position 35 of the AC from position 9 of the MQ. The

characteristic in the AC is reduced by 1 for each position shifted.* If the signs of the MQ and AC are different, the magnitude of the fraction in the AC is reduced by 1 before the shifting is begun. Each bit entering position 35 of the AC from position 9 of the MQ is inverted.

10. The MQ is given a characteristic which is 27 less than the characteristic in the AC, unless the AC contains a normal zero, in which case zeros are placed in positions 1-8 of the MQ.*

11. If the signs of the MQ and AC are different, the magnitude of the fraction in the AC is increased by 1. If a carry occurs between positions 8 and 9, the $C(AC)_{9\text{-}35}$ are shifted right one place and a one is inserted in $C(AC)_9$. If the carry from 9 to 8 occurs, the characteristic of the AC is increased by 1.

> *During execution of a floating-point addition, the AC or MQ overflow indicator and the corresponding light on the operator's console are turned on by too large a characteristic (overflow-characteristic greater than 255) or too small a characteristic (underflow-characteristic negative) in the AC or the MQ, respectively.

Unnormalized Floating Add
6-II UFA Y —0300

Same as floating add except steps 9a, 9b and 11 are omitted. No test is made for a normal zero in step 10.

Floating Subtract
7-II FSB Y +0302

Same as floating add except that step 2 is replaced by the following: the negative of the C(Y) is placed in the storage register.

Unnormalized Floating Subtract
7-II UFS Y —0302

Same as floating subtract except that steps 9a, 9b and 11 are omitted. No test is made for a normal zero in step 10.

Floating Multiply
17 FMP Y +0260

The C(Y) are multiplied by the C(MQ). The most significant part of the product appears in the AC and the least significant part appears in the MQ.

The product of two floating-point numbers is in normalized form if the multiplier and multiplicand are in this form. If either the multiplier or multiplicand is not in normalized form, the product is in normalized form only if the shift of one place in step 4b is sufficient to normalize it.

Floating-point multiplication takes place as follows:

1. The C(Y) are placed in the storage register and the AC is cleared.

2. The sum of the characteristics in the SR and in the MQ minus 128 is placed in positions 1-8 of the AC.*

3. The $C(SR)_{S,9\text{-}35}$ are algebraically multiplied by the $C(MQ)_{S,9\text{-}35}$. The most significant 27 bits plus sign of the product replace the $C(AC)_{S,9\text{-}35}$ and the least significant 27 bits replace the $C(MQ)_{9\text{-}35}$.

4a. If the fraction in the AC is zero, the $C(AC)_{Q,P,1\text{-}35}$ are cleared, yielding a normal zero. The sign of the AC is the algebraic sign of the product.

4b. If position 9 of the AC contains a zero but the fraction in the AC is not zero, the $C(AC)_{9\text{-}35}$ and the $C(MQ)_{9\text{-}35}$ are shifted left one position and the characteristic in the AC is reduced by 1.* The bit in position 9 of the MQ enters position 35 of the AC.

5a. If the AC contains a normal zero, positions 1-8 of the MQ are cleared.

5b. If the AC does not contain a normal zero, the $C(MQ)_{1\text{-}8}$ are replaced by a characteristic which is 27 less than the characteristic in the AC.*

6. The sign of the MQ is replaced by the same sign as that of the AC.

> *During execution of floating-point multiplication, too large or too small a characteristic in the AC or the MQ, respectively, turns on the AC or the MQ overflow indicator and the corresponding light on the operator's console.

Unnormalized Floating Multiply
17 UFM Y —0260

This operation is the same as floating multiply except that steps 4a, 4b and 5a are omitted.

Floating Divide or Halt
18-IV FDH Y +0240

The C(AC) are divided by the C(Y), the quotient appears in the MQ and the remainder appears in the AC. The MQ is cleared before actual division takes place.

If positions Q or P of the AC are not zero, division may take place and either or both of the AC and/or MQ overflow indicators may be turned on. When division by zero is attempted, the divide-check

indicator and light are turned on and the calculator stops, leaving the dividend in the AC unchanged. The quotient is in normalized form if both divisor and dividend are in that form. If divisor or dividend or both are not in normalized form, the quotient is in normalized form if

$$2\,|\,C(Y)_{9\text{-}35}\,| > |\,C(AC)_{9\text{-}35}\,| \geqq \tfrac{1}{2}\,|\,C(Y)_{9\text{-}35}\,|$$

Floating-point division takes place as follows:

1. The C(Y) are placed in the storage register.
2. If the magnitude of the fraction in the AC is greater than (or equal to) twice the magnitude of the fraction in the SR, the divide-check indicator and light are turned on, the calculator stops, and the dividend is left unchanged in the AC.
3. If the fraction in the AC is zero, the $C(MQ)_{1\text{-}35}$ and $C(AC)_{Q,P,1\text{-}35}$ are cleared and the remaining steps are skipped. The sign of the MQ is the algebraic sign of the quotient. The sign of the AC is the sign of the dividend.
4. If the magnitude of the fraction in the AC is greater than or equal to the magnitude of the fraction in the SR (but less than twice the magnitude of this fraction), the fraction in the AC is shifted right one position and the characteristic in the AC is increased by 1.* The bit in position 35 of the AC enters position 9 of the MQ.
5. The characteristic of the AC minus the characteristic of the SR plus 128 is placed in positions 1-8 of the MQ.*
6. The fractional part of the dividend, which consists of the $C(AC)_{S,9\text{-}35}$ (and the $C(MQ)_9$ if the condition of step 4 is met), algebraically divided by the fraction in the SR replaces the $C(MQ)_{S,9\text{-}35}$.
7. The 27-bit remainder resulting from the division in step 6 replaces the $C(AC)_{9\text{-}35}$. The sign of the AC is unchanged (i.e., the sign of the remainder agrees with the sign of the dividend.)
8. The characteristic in the AC is reduced by 27.*

   *During execution of a floating-point division, the AC or MQ overflow indicator and the corresponding light on the operator's console are turned on for too large or too small a characteristic in the AC or MQ, respectively.

Floating Divide or Proceed
18-IV FDP Y +0241

This operation is the same as floating divide or halt except for division by zero and step 2.

When division by zero is attempted, the divide-check indicator and light are turned on, division does not take place and the calculator proceeds to the next instruction. If the magnitude of the fraction in the AC is greater than (or equal to) twice the magnitude of the fraction in the SR, the divide-check indicator and light are turned on, division does not take place and the calculator proceeds to the next instruction. The dividend in the AC is unchanged.

Determination of Overflow and Underflow

For the instructions FAD, FSB, UFA, UFS, FMP*, UFM the conditions are as follows:

Ov(AC)	Ov(MQ)	$C(AC)_Q$	Ac	Mq
On	Off		Overflow	OK
Off	On		OK	Underflow
On**	On	0	Overflow	Overflow
On***	On	1	Underflow	Underflow

*The AC and MQ overflow indicators are not turned on if the result is a normal zero.
**Impossible with FAD, UFA, FSB, UFS.
***Impossible with UFA, UFS.

For the floating point divide instructions FDP*, FDH, the conditions are:

Ov(AC)	Ov(MQ)	$C(MQ)_{1\text{-}8}$	Ac	Mq
On	Off		Underflow	OK
On	On		Underflow	Underflow
Off	On	129-255	OK	Underflow
Off	On	0-128	OK	Overflow

*The AC and MQ overflow indicators are not turned on if the result is a normal zero.

Control Operations

No Operation
2 NOP +0761

The calculator takes the next instruction in sequence.

Halt and Proceed
2 HPR +0420

This instruction causes the calculator to stop. If the start key on the operator's console is depressed, the calculator proceeds to the next instruction in sequence.

Enter Trapping Mode
2 ETM +0760...007

This instruction turns on the trapping indicator and also the trap light on the operator's console. The

calculator operates in the trapping mode until a leave trapping mode operation is executed or until either the clear or reset key is pressed on the console.

Leave Trapping Mode
2 LTM —0760 . . . 007

This instruction turns off the trapping indicator and the trap light on the operator's console. The calculator will not operate in the trapping mode until another enter trapping mode operation is executed.

NOTE: When the calculator is operating in the trapping mode, the location of every transfer instruction (except trap transfer instructions) replaces the address part of location 0000, whether or not the conditions for transfer of control are met. If the condition is met, the calculator takes the next instruction from location 0001 and proceeds from that point. The location of each transfer instruction replaces the address part of location 0000.

Halt and Transfer
2 HTR Y +0000

This instruction stops the calculator. When the start key on the operator's console is depressed, the calculator starts again, taking the next instruction from location Y and proceeding from there.

When the calculator stops, the effective address of the HTR instruction is placed in the instruction location counter before executing any instruction. If TSX is manually executed, the 2's complement of this effective address is placed in the specified index register, and the transfer is executed.

Transfer
2 TRA Y +0020

This instruction causes the calculator to take its next instruction from location Y, and to proceed from there.

Transfer on Zero
2 TZE Y +0100

If the $C(AC)_{Q,P,1\text{-}35}$ are zero, the calculator takes its next instruction from location Y and proceeds from there. If they are not zero, the calculator proceeds to the next instruction in sequence.

Transfer on No Zero
2 TNZ Y —0100

If the $C(AC)_{Q,P,1\text{-}35}$ are not zero, the calculator takes its next instruction from location Y and proceeds from there. If they are zero, the calculator proceeds to the next instruction in sequence.

Transfer on Plus
2 TPL Y +0120

If the sign bit of the AC is positive, the calculator takes the next instruction from location Y and proceeds from there. If the sign bit of the AC is negative, the calculator proceeds to the next instruction in sequence.

Transfer on Minus
2 TMI Y —0120

In the sign bit of the AC is negative, the calculator takes the next instruction from location Y and proceeds from there. If the sign bit of the AC is positive, the calculator proceeds to the next instruction in sequence.

Transfer on Overflow
2 TOV Y +0140

If the AC overflow indicator and light are on as the result of a previous operation, the indicator and light are turned off and the calculator takes the next instruction from location Y and proceeds from there. If the indicator and light are off, the calculator proceeds to the next instruction in sequence.

Transfer on No Overflow
2 TNO Y —0140

If the AC overflow indicator and light are off, the calculator takes the next instruction from location Y and proceeds from there. If the indicator and light are on, the calculator proceeds to the next instruction in sequence after turning off the indicator and light.

Transfer on MQ Plus
2 TQP Y +0162

If the sign bit of the MQ is positive, the calculator takes the next instruction from location Y and proceeds from there. If the sign bit of the MQ is negative, the calculator proceeds to the next instruction in sequence.

Transfer on MQ Overflow
2 TQO Y +0161

If the MQ overflow indicator and light have been turned on by an overflow or underflow in the MQ

characteristic during a previous floating-point operation, the indicator and light are turned off, the calculator takes the next instruction from location Y and proceeds from there. If the indicator and light are not on, the calculator proceeds to the next instruction in sequence.

Transfer on Low MQ
2 TLQ Y +0040

If the C(MQ) are algebraically less than the C(AC), the calculator takes the next instruction from location Y and proceeds from there. If the C(MQ) are algebraically greater than or equal to the C(AC), the calculator proceeds to the next instruction in sequence.

Transfer and Set Index
2 TSX Y +0074

Not indexable. This instruction places the 2's complement of the location of this instruction in the specified index register. The calculator takes the next instruction from location Y and proceeds from there.

The 2's complement is used in this instruction because indexing is a subtractive process on the 704 and subtracting the 2's complement of a number is equivalent to adding the number.

Transfer with Index Incremented
2 TXI Y +1000

Not indexable. Contains a decrement part. This instruction adds the decrement to the number in the specified index register and replaces the number in the index register with this sum. The calculator takes the next instruction from location Y and proceeds from there.

Transfer on Index High
2 TXH Y +3000

Not indexable. Contains a decrement part. If the number in the specified index register is greater than the decrement, the calculator takes the next instruction from location Y and proceeds from there.

If the number in the specified index register is less than or equal to the decrement, the calculator proceeds to the next instruction in sequence.

Transfer on Index Low or Equal
2 TXL Y —3000

Not indexable. Contains a decrement part. If the number in the specified index register is less than or equal to the decrement, the calculator takes the next instruction from location Y and proceeds from there.

If the number in the specified index register is greater than the decrement, the calculator proceeds to the next instruction in sequence.

Transfer on Index
2 TIX Y +2000

Not indexable. Contains a decrement part. If the number in the specified index register is greater than the decrement, the number in the index register is reduced by the amount of the decrement and the calculator takes the next instruction from location Y and proceeds from there.

If the number in the specified index register is equal to or less than the decrement, the number in the index register is unchanged and the calculator proceeds to the next instruction in sequence.

Transfer on No Index
2 TNX Y —2000

Not indexable. Contains a decrement part. If the number in the specified index register is equal to or less than the decrement, the number in the index register is unchanged, the calculator takes the next instruction from location Y and proceeds from there.

If the number in the specified index register is greater than the decrement, the number in the index register is reduced by the amount of the decrement and the calculator proceeds to the next instruction in sequence.

Trap Transfer
2 TTR Y +0021

This instruction causes the calculator to take its next instruction from location Y and to proceed from there *whether in the trapping mode or not.* This makes it possible to have an ordinary transfer even when in the trapping mode.

P Bit Test
2 PBT —0760 . . . 001

If the $C(AC)_P$ is a one, the calculator skips the next instruction and proceeds from there. If position P contains a zero, the calculator takes the next instruction in sequence.

Low Order Bit Test
2 LBT +0760 . . . 001

If the $C(AC)_{35}$ is a one, the calculator skips the next instruction and proceeds from there. If position 35 contains a zero, the calculator takes the next instruction in sequence.

Divide Check Test
2 DCT +0760 . . . 012

If the divide-check indicator and light are on, the indicator and light are turned off, and the calculator takes the next instruction in sequence. If the indicator and light are off, the calculator skips the next instruction and proceeds from there.

Redundancy Tape Test
2 RTT —0760 . . . 012

If the tape-check indicator and light are on, the indicator and light are turned off and the calculator takes the next instruction in sequence. If the indicator and light are off, the calculator skips the next instruction and proceeds from there.

Compare Accumulator with Storage
3 CAS Y +0340

If the C(Y) are algebraically less than the C(AC), the calculator takes the next instruction in sequence. If the C(Y) are algebraically equal to the C(AC), the calculator skips the next instruction and proceeds from there. If the C(Y) are algebraically greater than the C(AC), the calculator skips the next two instructions and proceeds from there. Two numbers are algebraically equal when the magnitude of the numbers and the sign are both equal. A plus zero is algebraically larger than a minus zero.

Indexing Operations

Load Index from Address
2 LXA Y +0534

Not indexable. The address part of the C(Y) replaces the number in the specified index register. The C(Y) are unchanged.

Load Index from Decrement
2 LXD Y —0534

Not indexable. The decrement part of the C(Y) replaces the number in the specified index register. The C(Y) are unchanged.

Store Index in Decrement
2 SXD Y —0634

Not indexable. The $C(Y)_{3\text{-}17}$ are cleared and the number in the specified index register replaces the decrement part of the C(Y). The $C(Y)_{S,1,2,18\text{-}35}$ are unchanged. The contents of the index register are unchanged if one index register is specified. If a multiple tag is specified, the "logical or" of the contents of these index registers will replace the $C(Y)_{3\text{-}17}$ and will also replace the contents of the specified index registers.

Place Address in Index
2 PAX +0734

Not indexable. The address part of the C(AC) replaces the number in the specified index register. The C(AC) are unchanged.

Place Decrement in Index
2 PDX —0734

Not indexable. The decrement part of the C(AC) replaces the number in the specified index register. The C(AC) are unchanged.

Place Index in Decrement
2 PXD —0754

Not indexable. The AC is cleared and the number in the specified index register is placed in the decrement part of the AC. The contents of the index register are unchanged if one index register is specified. If a multiple tag is specified, the "logical or" of the contents of these index registers will replace the $C(AC)_{3\text{-}17}$ and will also replace the contents of the specified index registers.

Input-Output Operations

The identifying numbers for the various input-output components appear in the address part of an instruction whenever the programmer wants to operate one of these units. Whether the address part of an instruction refers to a storage location or to one of the components depends on the operation part of the instruction. Some operations make no sense if the address is interpreted as a location in storage; other operations make no sense if the address is interpreted as a component identification. Thus, an address is automatically interpreted by the calculator in the light of what it is asked to do by the operation part of the instruction.

The addresses of the input-output units are given below.

COMPONENT	OCTAL	DECIMAL
CRT	030	024
Tapes		
Binary Coded Decimal	201-212	129-138
Binary	221-232	145-154
Drum	301-310	193-200
Card Reader	321	209
Card Punch	341	225
Printer	361	241

Read Select
2-III RDS Y +0762

This instruction causes the calculator to prepare to read one record of information from the component specified by Y. If Y specifies a tape unit, the MQ is cleared by this instruction.

Write Select
2-III WRS Y +0766

This instruction causes the calculator to prepare to write one record of information on the component specified by Y. WRS 333_8 is used to *delay* the execution of any instruction until the MQ is available for computing after reading information from a tape.

Backspace Tape
2-III BST Y +0764

This instruction causes the tape designated by Y to space one record in a backward direction. If the tapes designated by Y is positioned at the load point, the BST Y instruction is interpreted as no operation.

Write End of File
2-III WEF Y +0770

This instruction causes the tape unit designated by Y to leave an end-of-file space, an end-of-file mark and a redundancy character on its tape.

Rewind
40ms-III REW Y +0772

This instruction causes the tape unit designated by Y to rewind its tape to the load point.

End of Tape Test
2 ETT —0760 . . . 011

This instruction must be given while the tape unit is selected (i.e., after a WRS, RDS, or WEF instruction and before the tape disconnects; no more than 744 μsec after the last CPY if WRS instruction; no more than 420 μsec after reading the last word, if RDS; and anytime up to 40 ms, if WEF). Failure to program this instruction may cause the tape to be pulled from its reel. If the tape indicator and the tape indicator light are off, the calculator skips the instruction immediately following the ETT and proceeds from that point. If the tape indicator and the tape indicator light are on, they will be turned off and the calculator will take the next instruction in sequence (no skip).

If tape instructions are given to a tape while the tape indicator is on, they will operate normally.

Locate Drum Address
2 LDA Y +0460

This instruction follows a read select or write select instruction referring to a drum unit and the address part of the C(Y) specifies the first location of the record to be read from or written on the drum. Not giving this instruction is equivalent to giving the instruction with the address part of the C(Y) equal to zero.

Copy and Skip
—III CPY Y +0700

This instruction is used following an RDS, WRS, or another CPY instruction to transfer a word of information between location Y in storage and an input-output component specified by the address part of the preceding RDS or WRS instruction. When this instruction is executed, the 36-bit word is formed in the MQ and then transmitted to storage or to the component. If the CPY instructions are not given within specific time ranges (found in the descriptions of these components), the calculator stops and a read-write check light on the operator's console is turned on.

If an additional CPY instruction is given after the last word of a unit record has been copied from a card or a record of tape, the CPY is not executed and the calculator skips the two instructions immediately following the CPY and proceeds from there. If an additional RDS instruction is given for which there is no corresponding record, the calculator sets up an end-of-file condition. The first CPY instruction given after this RDS is not executed; instead, the calculator skips the instruction immediately following the CPY and proceeds from there.

Plus Sense
2 PSE Y +0760

This instruction provides a means of testing the status of sense switches (and of turning on or off the sense lights on the operator's console), thus providing the programmer with flexible means of altering the sequence of instructions being executed. This instruction also permits the transmission of an impulse to or from the exit or entry hubs on the printer or card punch.

The address part of this instruction determines whether a light, switch, printer, or card punch is being sensed, and it further determines which light, switch, or hub is being sensed. The octal addresses for the different sense instructions are:

140	Turn off all sense lights on the operator's console.
141-144	Turn on sense light 1, 2, 3, or 4, respectively, on the operator's console.
161-166	If the corresponding sense switch on the operator's console is down (on), the calculator skips the next instruction and proceeds from there. If the sense switch is up (off), the calculator takes the next instruction in sequence.
341-342	The calculator causes an impulse to appear at the specified exit hub of the punch control panel.
360	If an impulse is present on the entry hub of the printer control panel, the calculator skips the next instruction and proceeds from there. If there is no impulse, the calculator takes the next instruction in sequence.
361-372	The calculator causes an impulse to appear at the specified exit hub of the printer control panel.

Minus Sense
2 MSE Y —0760

This instruction provides a means of testing the status of sense lights on the operator's console. The addresses of the four sense lights are:

141-144	If the corresponding sense light is on, the light is turned off, the calculator skips the next instruction and proceeds from there. If the light is off, the calculator takes the next instruction in sequence.

INSTRUCTION TIMING

Timing

The time required for the execution of any instruction is an integral multiple of the fundamental Most operations require a fixed number of cycles. The four types of exceptions are denoted by this fixed number supplemented by a Roman numeral I, II, III or IV, in the section "Instructions" and in Table I.

The four types of exceptions are as follows:

Type I: The instruction will be executed in two cycles if the extent of shift is nine places or less. Each additional 12 places of shift, or portion thereof, requires another cycle.

Type II: The instruction will be executed in seven cycles if the extent of shift is ten places or less in step 5 of FAD, UFA, FSB, and UFS and, also, if the extent of shift is four places or less in step 9b of FAD and FSB.

In step 5, each additional twelve places of shift, or portion thereof, requires another cycle.

In step 9b, each additional four places of shift, or portion thereof, requires another cycle.

Type III: The execution of this instruction may be delayed an indefinite length of time after its interpretation, depending on the status of the input-output components. For example, if two RDS instructions are given in succession for the same tape, the execution of the second RDS instruction will be delayed until the first record has been passed over. When a CPY instruction is given, the electronic and mechanical equipment must be synchronized and short delays may result while this synchronization takes place. In general, any execution delays of this type are of varying lengths depending on the programming.

Type IV: The execution of a floating divide instruction requires only three cycles if the fraction of the dividend is zero.

OPERATION	CYCLES	ALPHA CODE	OCTAL CODE
Halt and Transfer	2	HTR	+0000
Transfer	2	TRA	+0020
Trap Transfer	2	TTR	+0021
Transfer on Low MQ	2	TLQ	+0040
Transfer and Set Index*	2	TSX	+0074
Transfer on Zero	2	TZE	+0100
Transfer on No Zero	2	TNZ	—0100
Transfer on Plus	2	TPL	+0120
Transfer on Minus	2	TMI	—0120
Transfer on Overflow	2	TOV	+0140
Transfer on No Overflow	2	TNO	—0140
Transfer on MQ Overflow	2	TQO	+0161
Transfer on MQ Plus	2	TQP	+0162
Multiply	20	MPY	+0200
Multiply and Round	20	MPR	—0200
Divide or Halt	20	DVH	+0220
Divide or Proceed	20	DVP	+0221
Floating Divide or Halt	18 IV	FDH	+0240
Floating Divide or Proceed	18 IV	FDP	+0241
Floating Multiply	17	FMP	+0260
Unnormalized Floating Multiply	17	UFM	—0260
Floating Add	7 II	FAD	+0300
Unnormalized Floating Add	6 II	UFA	—0300
Floating Subtract	7 II	FSB	+0302
Unnormalized Floating Subtract	7 II	UFS	—0302
AND to Storage	4	ANS	+0320
AND to Accumulator	3	ANA	—0320
Compare Accumulator with Storage	3	CAS	+0340
Add and Carry Logical Word	2	ACL	+0361
Add	2	ADD	+0400
Subtract Magnitude	2	SBM	—0400
Add Magnitude	2	ADM	+0401
Subtract	2	SUB	+0402
Halt and Proceed	2	HPR	+0420
Locate Drum Address	2	LDA	+0460
Clear and Add	2	CLA	+0500
Clear and Add Logical Word	2	CAL	—0500
OR to Accumulator	2	ORA	—0501
Clear and Subtract	2	CLS	+0502
Load Index from Address*	2	LXA	+0534
Load Index from Decrement*	2	LXD	—0534
Load MQ	2	LDQ	+0560
Store MQ	2	STQ	—0600

* Not indexable.
** Not indexable but contains a decrement part.

OPERATION	CYCLES	ALPHA CODE	OCTAL CODE
Store	2	STO	+0601
Store Logical Word	2	SLW	+0602
OR to Storage	2	ORS	—0602
Store Left Half MQ	2	SLQ	—0620
Store Address	2	STA	+0621
Store Decrement	2	STD	+0622
Store Prefix	2	STP	+0630
Store Index in Decrement*	2	SXD	—0634
Copy and Skip	– III	CPY	+0700
Place Address in Index*	2	PAX	+0734
Place Decrement in Index*	2	PDX	—0734
Place Index in Decrement*	2	PXD	—0754
Plus Sense	2	PSE	+0760
Minus Sense	2	MSE	—0760
Clear Magnitude	2	CLM	+0760...000
Low Order Bit Test	2	LBT	+0760...001
P Bit Test	2	PBT	—0760...001
Change Sign	2	CHS	+0760...002
Set Sign Plus	2	SSP	+0760...003
Set Sign Minus	2	SSM	—0760...003
Complement Magnitude	2	COM	+0760...006
Enter Trapping Mode	2	ETM	+0760...007
Leave Trapping Mode	2	LTM	—0760...007
Round	2	RND	+0760...010
End of Tape Test	2	ETT	—0760...011
Divide Check Test	2	DCT	+0760...012
Redundancy Tape Test	2	RTT	—0760...012
No Operation	2	NOP	+0761
Read Select	2 III	RDS	+0762
Backspace Tape	2 III	BST	+0764
Write Select	2 III	WRS	+0766
Write End of File	2 III	WEF	+0770
Rewind	40 ms– III	REW	+0772
Long Left Shift	2 I	LLS	+0763
Logical Left	2 I	LGL	—0763
Long Right Shift	2 I	LRS	+0765
Accumulator Left Shift	2 I	ALS	+0767
Accumulator Right Shift	2 I	ARS	+0771
Rotate MQ Left	2 I	RQL	—0773
Transfer with Index Incremented**	2	TXI	+1000
Transfer on Index**	2	TIX	+2000
Transfer on No Index**	2	TNX	—2000
Transfer on Index High**	2	TXH	+3000
Transfer on Index Low or Equal**	2	TXL	—3000

TABLE I

A DETAILED description of each of the Type 704 components will be found in this section.

MAGNETIC TAPE UNITS

IN ADDITION to magnetic core and magnetic drum storage, ten Type 727 tape units with an associated control unit are available on the 704. These tape units are compatible with the tape units used on the Types 702 and 705 EDPM.

Each tape unit may contain a half-inch-wide oxide-coated plastic tape up to 2400 feet long on which information is stored as bits in the form of magnetized spots. The mechanism (read-write head) that reads or writes information on the tape is preceded by an erase head which erases the tape prior to writing, but *not* while reading. Hence, the same tape may be re-used many times by writing new information on it.

The reading, writing, and backspacing speed of the tapes is 75 inches per second. The longitudinal density of the tapes is 200 bits per inch. Reading or writing is done at the rate of 2500 words per second after the tapes are placed in motion. Tapes are read or written in a forward direction only; but the same tape may be written, backspaced, read, backspaced and written again in that order. Thus a record may be written and then read for checking purposes before writing the succeeding record.

The normal rewinding speed of the tapes is 75 inches per second if the length of tape to be rewound does not exceed 450 feet. The tape unit automatically measures the length of tape to be rewound. The time for a high-speed rewind of a reel of tape of any length from 450 to 2400 feet is nearly constant (about 1.2 minutes, allowing for acceleration and deceleration time).

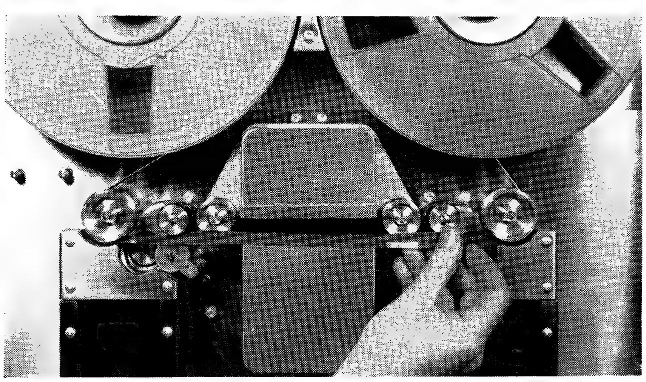
Reflective spots on the tape, made of adhesive aluminum stripping, are photo-electrically sensed to indicate the load point and the physical end of the tape, as indicated in Figure 18.

Operating Modes

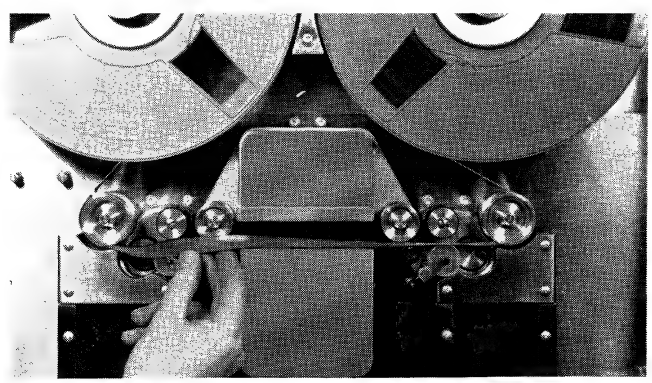
Peripheral equipment (card-to-tape, tape-to-card, and tape-to-printer) requires information to be stored as binary-coded decimal (BCD) characters. Therefore, the 704 operates in two distinct modes, depending on the address used to select the tape unit:

MODE	OCTAL ADDRESS	DECIMAL ADDRESS
BCD	201-212	129-138
Binary	221-232	145-154

When operating in the binary mode, the calculator reads or writes words without altering the bit pattern during transmission. When reading or writing in the BCD mode, the calculator alters the form of some of the BCD characters during transmission from or to



Load Point



Physical End

FIGURE 18

the tape. See "Character Alteration in BCD Mode."

Six bits make up one BCD character. Hence six BCD characters, comprising 36 bits, are transmitted with one copy and skip (CPY) instruction.

Physical Arrangement of Information on Tape

A ¾-inch blank space on the tape defines the *end of a record* of information. A 3.75-inch blank space, a tape mark followed by its redundancy character, and an end-of-record gap define the *end-of-file* of information (Figure 19). A tape may contain more than one file, and a file may contain any number of records. Each record contains an arbitrary number of words.

During a write operation, six bits and a redundancy check bit are recorded laterally across the tape. The lateral redundancy check bits are automatically placed on the tape to cause an even or odd number of binary 1's in each lateral row of tape for the BCD or binary mode, respectively. Also, at the end of each record written, a longitudinal redundancy check bit is placed automatically in each of the seven channels to cause an even number of binary 1's in each channel of that particular record. The longitudinal check is *always* an even check.

If information stored on a tape in the BCD mode is read in the binary mode, the tape-check indicator and corresponding light on the operator's console go on (because the lateral check bits are different), and the information is transmitted to storage *in unaltered form*. If a binary tape is read in the BCD mode, the tape-check indicator and light turn on, and the information is transmitted to storage in *altered form*.

In Figure 19, the tape is moving in the direction indicated by the arrows. Each y corresponds to the redundant bit for each six bits (x's) stored laterally, and each z corresponds to the redundant bit for each

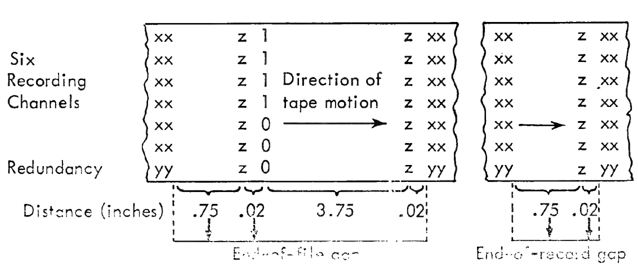


FIGURE 19

channel of the preceding record. The tape mark in the end-of-file gap has its own longitudinal check bits .020 inch beyond the tape mark. These check bits are identical to the tape mark—the special character 0001111_2.

Writing

The programming needed to write a record is write select (WRS) Y (Y denotes the tape unit and mode of checking), followed by a CPY instruction, to be repeated as many times as there are words in the record. This iterative procedure is known as a copy loop. After interpreting the first CPY, the calculator automatically delays its execution if the tape is not yet positioned to transmit the first word.

The WRS Y instruction starts in motion the tape designated by Y and selects the checking mode. If the copy loop is terminated, i.e., the calculator fails to receive a CPY within 336 microseconds (μs) of the preceding CPY, the calculator writes the longitudinal check bits and end-of-record gap and disconnects the tape unit. If another CPY is given after the tape is disconnected, the calculator will stop with the read-write check light turned on.

When a tape is written, the MQ cannot be used for computing between successive CPY instructions, or for 500 μs after the final CPY execution. The delay instruction, WRS 333_8 delays any instruction execution until the MQ is free.

Write End of File

The write end of file (WEF) causes the tape to erase an end-of-file gap and write a tape mark plus the corresponding longitudinal check bits. The calculator disconnects the tape immediately upon interpretation of the WEF instruction; hence, the MQ is free for computing while this instruction is executed. No tape instruction may be executed for 50 milliseconds following a WEF instruction.

To write more than one file of information, it is only necessary to write an end of file after writing the first file of information. At any later time, the first record of the second file of information can be written.

When a file, other than the last file, on a multi-file tape is rewritten, all succeeding files on the tape must be rewritten if they are to be read.

Reading

The execution of an RDS instruction starts the tape in motion, selects the checking mode, and clears the MQ. If the MQ is used for computing between the RDS and the first CPY, it must be cleared by the program before the first CPY instruction is given. After a CPY Y, during the reading loop, the word read into location Y in core storage is immediately available to the program. If a CPY is not given within 288 μs of the preceding CPY, the calculator disconnects the tape, which continues in motion until it reaches the end-of-record gap. Any input-output component other than a tape may be selected as soon as the calculator disconnects the tape. A select instruction, referring to tape, is not executed until the previously selected tape unit has been disconnected from the calculator. A CPY given after the calculator has disconnected the tape (with no other input-output unit selected) causes the calculator to stop with the read-write check light turned on.

If there are n words in a record being read, where n is unknown, the programmer may give CPY instructions until the calculator senses the end-of-record gap on the tape. CPY instruction $n + 1$ will not be executed; instead the calculator will skip *two* instructions following the CPY instruction and proceed from there. If an end-of-file gap is met, an RDS instruction must be executed to move the tape over the end-of-file gap and set up an end-of-file condition. The first CPY instruction given after the RDS is not executed; instead the calculator skips *one* instruction following the CPY and proceeds from there.

It is possible to bypass records (forward spacing) by executing RDS Y instructions without giving any CPY instructions. Because the lateral redundancy check is always effective on the first word of the record, whether it is transmitted to storage or not, the logical address of the tape unit must indicate the correct mode during tape spacing to avoid a tape-check indication. Each time a tape is selected by the RDS instruction and read or spaced over, the longitudinal redundancy information is automatically recalculated and compared with the redundancy information stored at the end of the record. If there is a discrepancy, the tape-check indicator and light go on and the on or off condition of seven neon lights on the tape control unit indicate the tape channels in error. If Y indicates a tape unit, another RDS Y instruction turns off the seven lights on the tape control unit, but the RDS does not turn off the tape check indicator.

It is possible to read the first n words in an N-word record where $n \leqq N$. When $n < N$, the nth word will be stored by the nth CPY, and the 704 will be disconnected after the $n + 1$ word appears in the MQ. The MQ may not be used between successive CPY instructions nor may it be used for 500 μs following the nth CPY.

Physical End of Tape

When the reflective spot, indicating the physical end of the tape, is reached during a write operation, the tape indicator and tape indicator light are turned on. There is no interruption to either the writing operation or subsequent calculator operations unless the program ignores the indication of the physical end of the tape and writing is continued, detaching the tape from the reel. The ETT instruction provides a means for the program to test the status of the tape indicator and to transfer to those instructions needed to terminate the file and rewind the tape reel. Because the ETT instruction turns off the tape indicator, the program should not attempt further writing on this tape because there will be no second indication that the end of the tape has been reached.

Backspacing

A backspace tape (BST) Y instruction spaces the tape one record backward. (Y can indicate either mode because no checking occurs when tape is backspaced.) When a BST Y is given where Y designates a tape in a rewound position, the calculator immediately disconnects the tape. The BST acts as a no operation instruction in this case. If the BST Y is given, where Y designates a tape that is positioned to read the first record of a second file, the tape is moved so that the read-write head is positioned 3/8″ in front of the tape mark. When the next operation on tape unit Y is a read instruction, the execution of a read in BCD mode differs from the execution of a read in the binary mode. In the BCD mode, the tape mark and its check character are recognized as an end-of-file and on the first CPY instruction the calculator skips one instruction and proceeds from there. In the binary mode, the tape mark, without the gap, is

read as a record of one character and no end-of-file indication is given. Executing another BST backspaces the tape over the last record in the first file. The MQ may be used for computing while the BST instruction is being executed.

The maximum number of times that BST followed by a write instruction is given is ten. Any number greater than ten may cause an end-of-file gap to be recorded.

Rewinding

The rewind (REW) Y instruction causes the tape designated by Y to return to its load point. If a REW is given while the tape is being read, the tape moves to the end of the record before the REW becomes effective. After the REW becomes effective, any input-output unit may be selected.

Testing Redundancy Information

The redundancy tape test (RTT) instruction tests the status of the tape-check indicator during tape reading or after information is read from a tape.

After the last character of a record has been copied into storage, 275 μs are required for the longitudinal redundancy check character to be interrogated in the tape control unit. Allowing for a 25 μs margin of safety, the RTT instruction tests both lateral and longitudinal information for the entire record if it is given 300 μs after the last word of the record is copied into storage. If the *delay* instruction, WRS 333_8, is given after the CPY is executed for the last complete word in the record and before the RTT is given, the calculator is allowed to complete the longitudinal tape checking before executing the RTT.

When not all of the words in a record are copied into storage and the remainder of the record is spaced over, the delay instruction will not effect the necessary delay for the longitudinal redundancy check to be completed, because the calculator disconnects the tape immediately when it fails to receive a CPY within the specified time requirement. The delay instruction requires ony 24 μs execution time after the tape is disconnected. To obtain a longitudinal check, therefore, the record spacing time must be used in computing before the RTT is given.

Timing

When an RDS or WRS is given, several milliseconds are required to start the tape in motion and position it to read or write the first word of the record. After its interpretation, the first CPY execution is automatically delayed an amount of time that is the difference between the time the CPY is given and the time the CPY is executed. See Table II.

Current Instruction	Subsequent to	First CPY Must Be Given within	First CPY Is Executed within: Aver.	Max.
WRS	REW	40 ms	50 ms	60 ms
WRS	WRS	7 ms	10 ms	12 ms*
WRS	RDS	7 ms	10 ms	12 ms
WRS	BST	7 ms	10 ms	12 ms
RDS	REW	20 ms	50 ms	60 ms
RDS	WRS	3 ms	10 ms	12 ms
RDS	RDS	3 ms	10 ms	12 ms*
RDS	BST	3 ms	10 ms	12 ms

*Some delay time may be used for computing if the tape is selected for reading or writing the next record before the tape stops moving. Thus, if t is the time between the last CPY of the preceding record and the WRS and if $t \leqq 3$ ms, then $10 - t$ ms can safely be used for computing before giving the first CPY of the next record. If t is the time between the last CPY of the preceding record and the RDS and if $t \leqq 2.5$ ms, then $8.5 - t$ ms can safely be used for computing before giving the first CPY of the next record.

TABLE II. Tape Timing

The calculator will execute all instructions other than input-output instructions 24 μs after the BST is executed. Thirty-six μs after the BST is executed, the calculator will execute all instructions other than *tape* input-output instructions.

The total time required to backspace the tape one record is computed by adding (1) the time required to start the tape moving in a backward direction, (2) the time required to space over the n words in the record, and (3) the time needed to stop and reposition the tape. The timing is given below.

CURRENT INSTRUCTION	SUBSEQUENT TO	TOTAL TIME IN MS START	MOVE	STOP
BST	WRS	43 ± 8.6	$.4n$	25
BST	RDS	30.5 ± 6.1	$.4n$	25
BST	BST	34.5 ± 6.9	$.4n$	25

If the tape is moving when the BST is given, the execution of the BST is delayed until the tape has stopped. This delay is 3 ms if the BST is given immediately after copying the last word of a record.

The WEF execution time is 50 ms. Any input-output unit other than tape may be used while the WEF is being executed.

Simultaneous Tape Writing

The following procedure can be used to write simultaneously on two or three tapes with logically distinct addresses if *all* tapes involved are rewound or if *all* tapes are not rewound.

Writing simultaneously on two tapes X_1 and X_2 (*octal addresses*):

Let X_2 equal 1, 2, . . . 12_8, specifying any one of the ten tapes.

Let X_2 equal 1, 2, . . . 12_8, specifying any one of the ten tapes.

BCD MODE	BINARY MODE
WRS $320 + X_1$	WRS $320 + X_1$
WRS $200 + X_2$	WRS $220 + X_2$
....	....
7 ms or less computation	7 ms or less computation
....	....
Copy loop	Copy loop

Writing simultaneously on three tapes X_1, X_2, X_3 (*octal addresses*):

Let X_1 equal 1, 2, . . . 12_8, specifying any one of the ten tapes.

Let X_2 equal 1, 2, . . . 12_8, specifying any one of the ten tapes.

Let X_3 equal 1, 2, . . . 12_8, specifying any one of the ten tapes.

BCD MODE	BINARY MODE
WRS $320 + X_1$	WRS $320 + X_1$
WRS $320 + X_2$	WRS $320 + X_2$
WRS $200 + X_3$	WRS $220 + X_3$
....	....
7 ms or less computation	7 ms or less computation
....	....
Copy loop	Copy loop

As many as three tapes can be written simultaneously by manually setting the rotary selector switches on the tape units to the same number and addressing the WRS to that number. Thus, if the number is 1, then WRS 201 writes information simultaneously in the BCD mode on the tapes whose selector switches are set to 1.

Incomplete Word on Tape

When a tape prepared by the 702 or 705 EDPM's or on the card-to-tape peripheral equipment is read, it is possible that some records do not have an integral multiple of six BCD characters. Note that a 36-bit word is transmitted from the MQ to core storage *only* when six groups of six bits each (six bits correspond to one character) are transmitted from the tape to the MQ.

When reading occurs in the BCD mode, the following procedure occurs automatically during the execution of a CPY:

1. If there are no more characters on the tape, the end-of-record skip takes place.
2. The altered character from the tape replaces the $C(MQ)_{S,1\text{-}5}$.
3. The C(MQ) are rotated left six places.
4. Step 1 (and possibly steps 2 and 3) is repeated until there are six characters in the MQ.
5. The C(MQ) are stored and the MQ is cleared. The calculator then proceeds to the next sequential instruction following the CPY instruction.

If less than six characters comprise the last word on tape, a CPY instruction cannot transmit them to storage. A CPY instruction given for the incomplete

word causes an end-of-record skip, leaving the incomplete word in the MQ. The tape check indicator will not be turned on because an incomplete word has entered the MQ. (*Note:* This will not decrease tape-checking when binary tapes are being read, because detectable incomplete binary words will be detected by the redundancy check.) If a CPY instruction is not given for the incomplete word, the extra characters are automatically brought into the MQ (because word $n + 1$ of a record is always transmitted to the MQ after n CPY's). A tape check occurs only if the computed lateral or longitudinal bits do not compare with those on the tape. Use the delay instruction, WRS 333_8 to delay the execution of any instruction until the MQ is available. After the delay instruction has been given, the store MQ (STQ) instruction may be used to store the contents of the MQ in core storage.

When an incomplete word is brought into the MQ from tape, the unused portion of the MQ contains zeros.

Character Alteration in BCD Mode

Altering characters when reading or writing in the BCD mode on the 704 changes the zones of some of the characters and the numerical code of the character representing zero. The zones differ from the 702 code because the 704 requires this zone change to help fast sorting procedures. Because redundancy checking is an even parity check on peripheral equipment (and in the BCD mode on the 704), the pure zero would not have a non-zero bit. Several pure zeros would correspond to an end-of-record gap. Thus, the zero character is changed to 00 1010 in the BCD mode. The zone alterations follow:

CLASS	IN 704	ON TAPE
Numerical	00	00
A to I	01	11
J to R	10	10
S to Z	11	01

Table III shows the automatic alteration of all characters during transmission in the BCD mode.

CHARACTER	IN STORAGE	ON TAPE	CHARACTER	IN STORAGE	ON TAPE
0	00 0000	00 1010	A	01 0001	11 0001
1	00 0001	00 0001	B	01 0010	11 0010
2	00 0010	00 0010	C	01 0011	11 0011
3	00 0011	00 0011	D	01 0100	11 0100
4	00 0100	00 0100	E	01 0101	11 0101
5	00 0101	00 0101	F	01 0110	11 0110
6	00 0110	00 0110	G	01 0111	11 0111
7	00 0111	00 0111	H	01 1000	11 1000
8	00 1000	00 1000	I	01 1001	11 1001
9	00 1001	00 1001	$\overset{+}{0}$	01 1010	11 1010
#	00 1011	00 1011	.	01 1011	11 1011
@	00 1100	00 1100	⌑	01 1100	11 1100
—	10 0000	10 0000	Blank	11 0000	01 0000
J	10 0001	10 0001	/	11 0001	01 0001
K	10 0010	10 0010	S	11 0010	01 0010
L	10 0011	10 0011	T	11 0011	01 0011
M	10 0100	10 0100	U	11 0100	01 0100
N	10 0101	10 0101	V	11 0101	01 0101
O	10 0110	10 0110	W	11 0110	01 0110
P	10 0111	10 0111	X	11 0111	01 0111
Q	10 1000	10 1000	Y	11 1000	01 1000
R	10 1001	10 1001	Z	11 1001	01 1001
$\overset{-}{0}$	10 1010	10 1010	‡	11 1010	01 1010
$	10 1011	10 1011	,	11 1011	01 1011
*	10 1100	10 1100	%	11 1100	01 1100
&	01 0000	11 0000			

TABLE III

Manual Operation of the Tape Units

On each tape unit, manual operations are performed by using the keys and lights appearing in Figure 20.

The rotary selector switch on a tape unit determines which one of the ten tape addresses may select this unit. If the switch is set to 1, the unit may be addressed by 201_8 in the BCD mode or 221_8 in the binary mode. Zero corresponds to the tenth tape unit.

The select light is turned on only when the calculator selects the tape unit. The tape unit is in ready status (the ready light is on), provided the tape is loaded into the columns, the reel door interlock is closed, and the tape unit is not in the process of finding the load point (rewind or load operation). Manual control is indicated when the ready light is off, provided the tape unit is not rewinding or loading and the reel door is closed.

Pressing the start key places the tape unit under control of the tape control unit (and, indirectly, the calculator) and causes the ready light to be turned on, provided the tape unit is in ready status. Pressing the reset key removes the tape unit from the calculator's control. It turns off the ready light, and resets all controls to their normal positions. It also stops any tape operation which has been initiated (except high-speed rewind, which will revert to low-speed rewind). After the tape is loaded into the vacuum columns and low-speed rewind is in progress, the reset key may be pressed again to stop the low-speed rewind.

When the door is open, the reel door interlock prevents operation of the reel drive motors. If the reel door is closed and the ready light is off, pressing the load-rewind key causes a fast rewind at the end of which the tape is loaded into the vacuum columns and searched in a backward direction for the load point. Pressing the unload key causes the tape unit to remove the tape from the vacuum columns and raise the head cover, regardless of the distribution of the tape on the two reels. If the tape is not at the load point when the operator wishes to change it, he starts a load point search by pressing the load-rewind key.

The tape indicator and the tape indicator light in a tape unit are turned on when the tape breaks or when the physical end of tape is reached during a write operation. The ETT may be used in a program to interrogate the status of the tape indicator in a selected tape unit. If the program selects the tape unit for reading or writing after the tape indicator is turned on, there will be no interruption to normal calculator operation.

The tape indicator and the tape indicator light may be turned off by pressing the reset key on the tape unit and then pressing the unload key on the tape unit. Execution of the ETT will turn off the tape indicator and the tape indicator light in a selected tape unit.

The plastic tape reels are 10½ inches in diameter. They are designed so that the front and back sides of the reel are different (Figure 21). In normal operation, a special ring is inserted in a groove in the

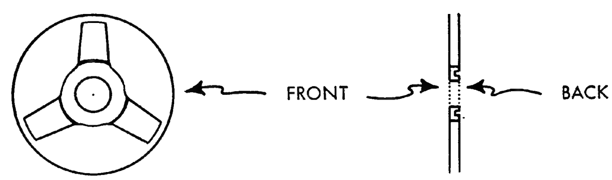


FIGURE 21

back side of the reel to depress a pin which is then under spring tension. If the special ring is removed from the reel, the pin rides freely in this groove and

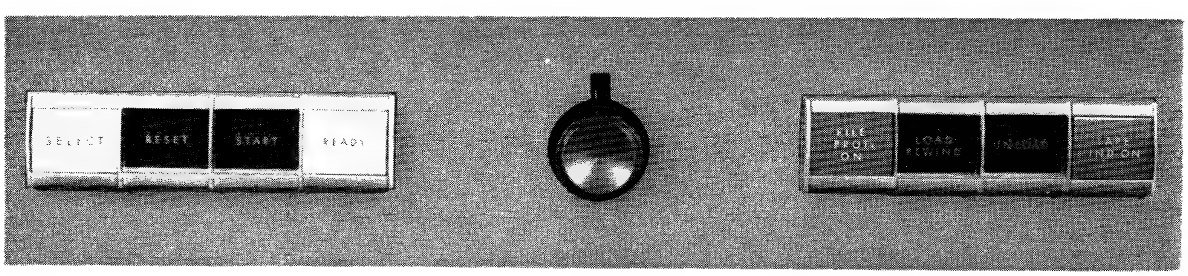


FIGURE 20

a writing interlock is automatically set. Also, the file protection light is turned on to inform the program that it is impossible for the program to write on the tape. However, this tape may be read, backspaced, or rewound freely when the file protection light is on.

MAGNETIC DRUMS

In addition to magnetic core and magnetic tape storage, two Type 733 magnetic drum units are available for the 704. Each magnetic drum unit has a storage capacity of 8192 words, each word consisting of 36 bits. A drum unit contains two distinct physical drums, each with a storage capacity of 4096 words.

Each physical drum consists of two logical drums whose octal addresses are indicated in Figure 22. Each logical drum has a storage capacity of 2048 words.

A logical drum is selected by giving the appropriate address 193-200 or 301-310 octal.

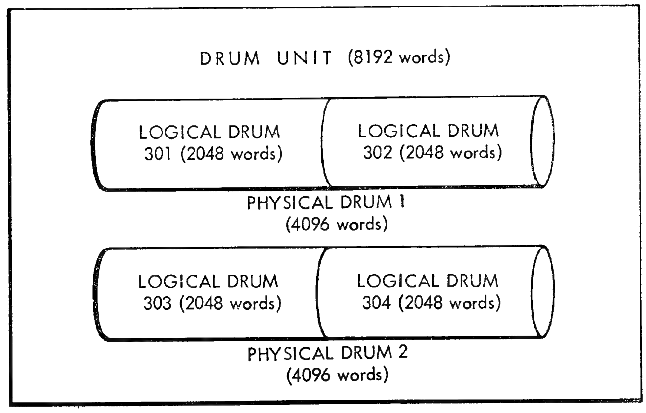


Figure 22

Physical Arrangement of Words on Drum

The 2048 locations on each logical drum can be individually addressed by integers in the range 0000-2047 decimal (0000-3777 octal). A record (block) of words is normally stored on a drum in sequentially numbered locations. The programmer must indicate the drum address where the first word of the core storage record is to be written on or read from the drum. The number of CPY instructions executed in the copy loop determines the number of words in the record.

Figure 23 illustrates the physical arrangement of words on a logical drum. The addresses are numbered octally. Observe that, when reading or writing a continuous record, the calculator refers to every eighth word of the drum for consecutive addresses.

Each logical drum has 256 sectors. Therefore, it must make eight complete revolutions for all 2048 words to be read or written as a continuous record.

Note: Drum sectors are numbered from 000 to 255 (000 to 377 octal). The eight least significant binary positions of the drum address of a word determine the number of a drum sector where a word is stored.

Reading and Writing

Because 96 μs are needed to read or write one word, successive words are written on or read from the drum at the rate of 10,000 words per second. A drum read select (RDS) Y or write select (WRS) Y selects one of the eight logical drums indicated by the address Y and connects it to the calculator. The drum then remains indefinitely selected waiting for a locate drum address (LDA) instruction. (If an LDA is not given, then the drum remains selected waiting for the first CPY.

The 11 least significant bits of the address part of the C(Y) in the LDA Y instruction specify the initial drum location of the record. If an LDA Y is not given, the first CPY refers to the drum address 0000. The automatic address counter for the drum has only 11 binary positions. Hence, if a 38-word record begins at 2040, the last word of the record is found at location 0029.

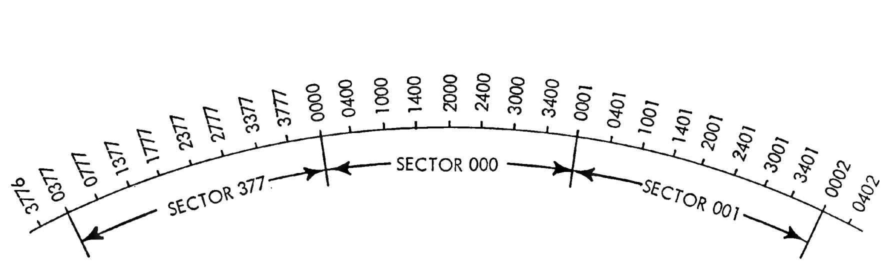


Figure 23

Following an LDA, the first CPY must be given within 36 μs (three cycles); otherwise, the drum may disconnect. The LDA is an indexable instruction.

When information is written on a drum, the execution of a CPY Y instruction causes the word at location Y in core storage to be loaded into the MQ from which it is transmitted to the drum. During a reading operation, the execution of a CPY Y instruction causes the word from the drum to be loaded into the MQ from which it is transmitted to location Y in core storage. The MQ cannot be used for computing during the copy loop.

Between successive CPY's, three cycles (36 μs) are available for programming (excluding the CPY itself). When a CPY is not given within three cycles of the preceding CPY, the drum disconnects. If a CPY is given after the drum has disconnected, the calculator stops with the read-write check light on.

Table IV shows the minimum time T between the execution of the RDS or WRS and the LDA (or first CPY if no LDA is given). During this entire time T, the calculator is available for computing. However, if any portion of time T is not used for calculating, the calculator delays the amount of time which is the difference between T and the time used for calculating during this period.

Drum Motion Time

The minimum time between the execution of the last copy of record x and the execution of the first CPY of record $x + 1$ is A, D, *or* $\bar{A}$ (Table IV). The average access time A is 12.29 ms, although it may be as high as 24 ms.

To compute D, subtract the final drum address in the preceding record from the initial drum address in the current record. If the result is negative, add 2048 to the result. Divide the result by 256; the quotient Q and the remainder R appear in the formula: $D = .012Q + .096R$. (This formula is used in computing the rotation time when the physical drum selected is the same as the last one used.)

To compute $\bar{A}$, divide the initial drum address by 256; the quotient Q' and remainder R' appear in the formula $D' = .012Q' + .096R'$. (This formula is used in computing the rotation time when the physical drum selected is different from the last one used.)

The computed rotation time is valid only when A, D, or $\bar{A} > T + .12$ ms.

Multiple Record

Because one drum revolution requires 24 ms, it is possible to read multiple records during a single revolution if the words to be read are stored on one physical drum in an optimal way. If the last CPY of the first record is followed immediately (within three cycles) by an RDS selecting the same physical drum, the LDA may be given for a drum address that is at least eight sectors beyond the drum address of the last word in the preceding record. Six sectors are passed over during execution of RDS and one sector is passed over during execution of LDA. An additional sector must be added for each 84 μs, or portion thereof, beyond the allowable 500 μs of programming between the RDS and the LDA instructions.

For an example, assume that a record is written on a drum where the last word of the record is stored in location 0200_{10}. We wish to know the earliest sector in which to place the first word of the next record so that both records can be read during the same drum

PREVIOUS INSTR.	CURRENT INSTR.	T (in ms)	ROTATION TIME (in ms)
RDS 301 or 302	RDS 301 or 302	0.5	D (or A if insufficient information is available to use the formula for D)
RDS 301 or 302	WRS 301 or 302	15.0	
WRS 301 or 302	RDS 301 or 302	15.0	
WRS 301 or 302	WRS 301 or 302	15.0	
RDS 301 or 302	RDS 303-310	0.5	$\bar{A} = A + D'$
RDS 301 or 302	WRS 303-310	15.0	$\bar{A} = A + D'$
WRS 301 or 302	RDS 303-310	15.0	$\bar{A} = A + D'$
WRS 301 or 302	WRS 303-310	15.0	$\bar{A} = A + D'$

TABLE IV

revolution when there are (a) 700 μs of programming between records, (b) 840 μs, (c) 500 μs or less.

(a) $700 \div 84 = 8+$
$= 9$ sectors for computing
1 sector to execute LDA
10 sectors to be skipped
Next record can begin at 0211_{10}.

(b) $840 \div 84 = 10$ sectors for computing
1 sector to execute LDA
11 sectors to be skipped
Next record can begin at 0212_{10}.

(c) The minimum sector allowance is seven sectors (this includes the sector necessary for the LDA).
Next record can begin at 0208_{10}.

PUNCHED CARDS

IN MOST applications magnetic tape is used as the principal input medium. It may be desirable to use IBM cards as an input medium in some situations, where the volume of input is sufficiently small to permit an economical operation. In either case, IBM cards are used as the medium for initially recording data because of their great flexibility and because of the availability of apparatus for key punching, verifying, and duplicating. Errors are easily detected and corrected, input data may be readily prepared on several key-punches simultaneously, and the cards may be collected before entry into the computer. Cards are particularly desirable when one wants to have manual access to a file. They can be easily separated. Their contents may be printed on them. It should be emphasized that the punched card input and output may represent any alphabetic character or special symbol, provided only that a program exists to recognize the IBM code for this information. A program may also provide for quantities to be represented in any number system and read or punched accordingly.

Entering a program on cards may be done in such a way that instructions are punched, one to a card, in the form most desirable to the programmer (e.g., in decimal notation). The computer can then be supplied with a standard program to assemble the instructions in the desired order. Then, if errors are detected or if changes must be made, the wrong cards are removed, the correct ones (not necessarily the same number of cards) are added, and the computer prepares the new program. Note that there is no need to repunch any but the cards in question.

The card-feeding mechanism in the card reader is similar to that in the Type 402 Accounting Machine and includes two sets of 80 reading brushes. Correspondingly, there are 80 punching magnets and 80 punching brushes in the card punch. Only 72 columns of the standard IBM card, however, can be read into core storage, and only 72 columns can be punched from core storage (unless split-column wiring is used). Any 72 columns of the card can be selected through control panel wiring. For simplicity in the following discussion, assume that columns 1 to 72 of the card are used for both reading and punching.

Binary information is represented on a card as follows: each of the 12 rows of the card is split into two parts, the left half consisting of columns 1 to 36 and the right half of columns 37 to 72; each half row can be treated as a 36-bit word and read into a location in core storage.

Figure 24 shows how the card is divided. In this particular example, the first 72 columns of the card are used. Each of the rows is split into half-rows of 36 columns each. Thus, the half-row identified by the circled 9 is named the 5-row left. Similarly, the row identified by the circled 10 is named the 5-row right. Thus, there are 24 half-rows in the card. One full word of binary information can be punched in any half-row (including sign). The machine regards any punched hole as a binary 1. "No punch" indicates a binary 0. Thus, an 8-punch in column 36 of the card is regarded by the machine as a binary 1 in the least significant position of the binary word punched in the 8-row left. The leftmost position of each half-row is reserved for the sign bit of the word. A binary 1 represents a negative sign, while a binary 0 represents a positive sign.

NOTE: The exact position of a word that each column represents is completely arbitrary according to how the particular control panel is wired.

Observe that this card representation of 24 binary words does not mean that the cards must always be punched with true binary information. The holes in the card can just as well be numerical punching in the standard decimal card code, alphabetic punching, or control punching. It is necessary only to provide a suitable program for the computer to translate between the binary code in which it operates and the

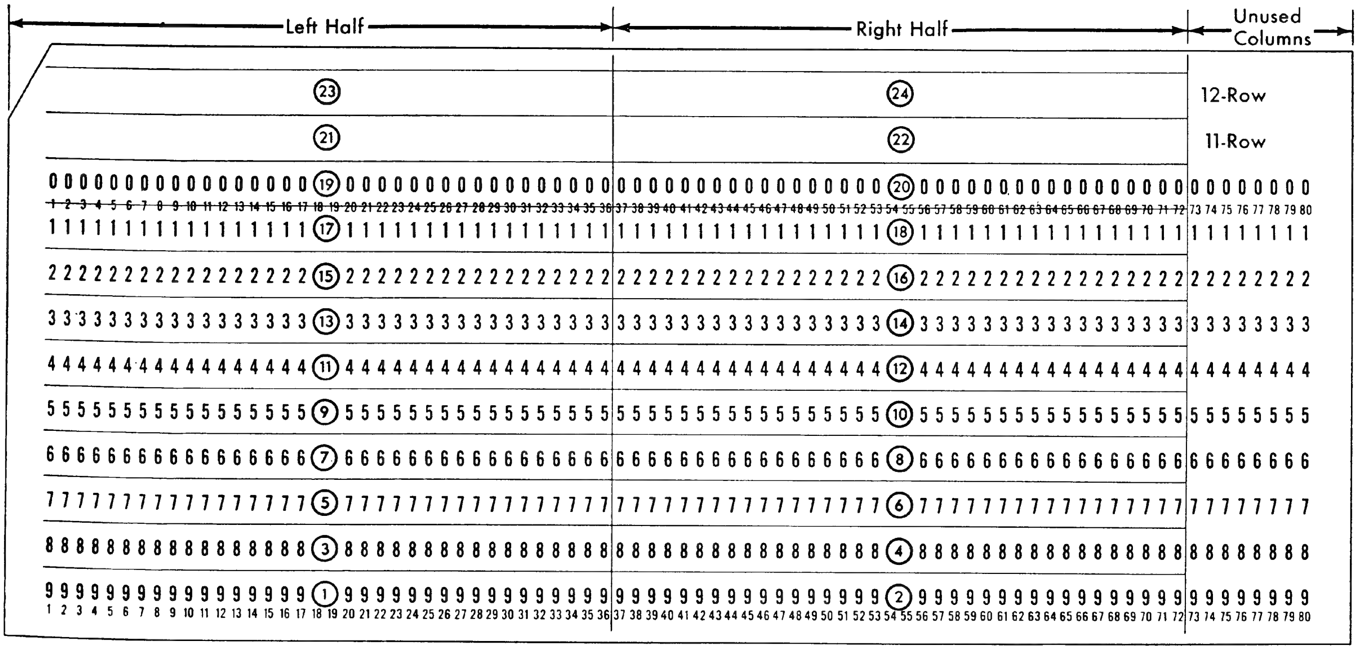


FIGURE 24

particular code used on the card. The translation to and from the decimal numerical code, for instance, can proceed simultaneously with reading and punching so that the over-all card-handling speed is not reduced below the standard rates of 150 or 250 cards per minute for reading and 100 cards per minute for punching.

Feed cards face down, 9's edge first, in *both* the card reader and card punch. The internal card circuits are arranged so that the 24 half-rows of the card are read or punched in the sequence indicated by the circled numbers in Figure 24. The sequence of reading or punching full words is then as follows: 9-row left, 9-row right, 8-row left, 8-row right, and so on to 12-row left, 12-row right.

For reading and punching cards, a unit record is defined as the information contained in *one* card. A file consists of any number of unit records. It takes the form of a deck of cards. Note that definitions of unit records and files are usually different, depending on the particular input or output component being discussed.

CARD READER

EITHER one of the two Type 711 card readers, model 1 or model 2, can be used on the 704. The model 1 reads cards at the rate of 150 cards per minute, the model 2 reads cards at the rate of 250 cards a minute. The principal difference is found in the timing section.

For a program to cause the calculator to read all of the information punched on a card into core storage, it is necessary to give an RDS instruction with an address of 209 (card-reader identification) followed by 24 CPY instructions.

The RDS instruction causes the card-feeding mechanism to start in motion. The program then is free to continue any operations until the 9-row of the card appears under the reading brushes. At this time, the program must provide a CPY Y which causes the word punched in the 9-row left to be read and stored in core storage location Y. The program can then resume until the calculator is prepared to read information punched in the 9-row right. The program now must supply another CPY instruction to read this word into core storage. This procedure continues until all 24 half-rows have been read. Because of their functions, these CPY instructions are called 9 left CPY, 9 right CPY, and so on. Another RDS must be given to read another unit record (card).

The RDS instruction can be given, followed immediately by the 24 CPY instructions in succession, without any other operations being done between instructions. In such a case the calculator waits automatic-

ally until a half-row is in position to be read before executing the CPY instruction.

The intervals of time between these instructions which may be used for useful calculating are definitely limited and are completely specified below. If a CPY is given *after* the card reader is in position to read a given half-row, the machine stops, and the read-write check light turns on at the operator's console. The amount of calculating time available between the last CPY instruction for a given card and the RDS instruction that initiates the reading of a succeeding card is unlimited. But if an RDS instruction does not occur within a definite time limit, the card reader stops. It will start up only after the new RDS instruction has been received. To keep the card reader in continuous motion and operating at its full speed of 150 or 250 cards per minute, the time limits discussed below must be observed.

Calculator operation is such that during execution of a CPY, the word read from a half-row of the card first enters the MQ before being sent to core storage. This, of course, destroys any information previously stored in the MQ.

If a 25th CPY instruction is given after an RDS instruction, the card reader will already have set up an end-of-record condition (denoting that all 24 half-rows of the card have been read). Under this condition, the 25th CPY is not executed, and the program skips to the *third* instruction after the CPY. In this way the program may transfer control to a section that will cause the succeeding card to be read.

When the hopper of the card reader becomes empty, the calculator stops. Depress the start key on the card reader to allow the cards remaining ahead of the reading station to be read under control of the program. After the last card has been read in this way, and if another RDS instruction followed by a CPY is given, the card reader sets up an end-of-file condition. Under this condition the CPY instruction is not executed, and the program skips to the *second* instruction following the CPY. In this way, for example, control may be transferred to a particular section of the program that continues a calculation interrupted by the card-reading procedure.

The contents of the 24 locations of core storage, into which the 24 half-rows have been read, is known as the *card image*. By a program that suitably manipulates this card image, decimal information punched in standard IBM code may be converted to binary information.

In reading cards it is not always necessary to follow an RDS by 24 CPY instructions. The card reader normally reads half-rows for every following CPY up to 24. If, however, after a few CPY instructions, another RDS is given, the card reader automatically ignores any succeeding half-rows that have not been read and starts reading a new card. Thus, for instance, it is possible to read the first five words of a card and ignore the rest. It is not possible, however, to read the first five words, skip the sixth and seventh words, and continue on reading the card. A CPY instruction designed to accomplish any reading of this type always results in a machine stop and a read-write check light. If successive RDS instructions are given with no intervening CPY instructions, the net result is the feeding of cards through the machine with no words being read into storage.

Timing for Model 1

Cards are read at the rate of 150 per minute. In continuous card reading, 292 ms of the card cycle of 400 ms are available for useful calculating. The difference, 108 ms, is required for the execution of the CPY and RDS instructions and appropriate time-margins for safe synchronization of mechanical and electronic components.

The maximum safe times available for computing between executions of CPY instructions are indicated in Figure 25. For example, after execution of the 9-left CPY, 540 μs are available before the 9-right CPY must be given; and after execution of the 9-right CPY, 15 ms are available before the 8-left CPY execution. The RDS must be given in the hatched portion for continuous operation of the card reader. After the 12-right CPY execution of a card, however, it takes 20 ms before the machine can execute an RDS for the next card. If the RDS then, is given *t* ms after the 12-right CPY, and if *t* is less than 20, the machine will compute (i.e., it will proceed with any intervening programs) for these *t* ms. Upon receiving the RDS, however, the program will be delayed until 20 ms (from the 12-right CPY) have elapsed. If the RDS is given after the interval of 20 ms, the program will not be delayed. So to be able to compute for *all* of the available time between cards,

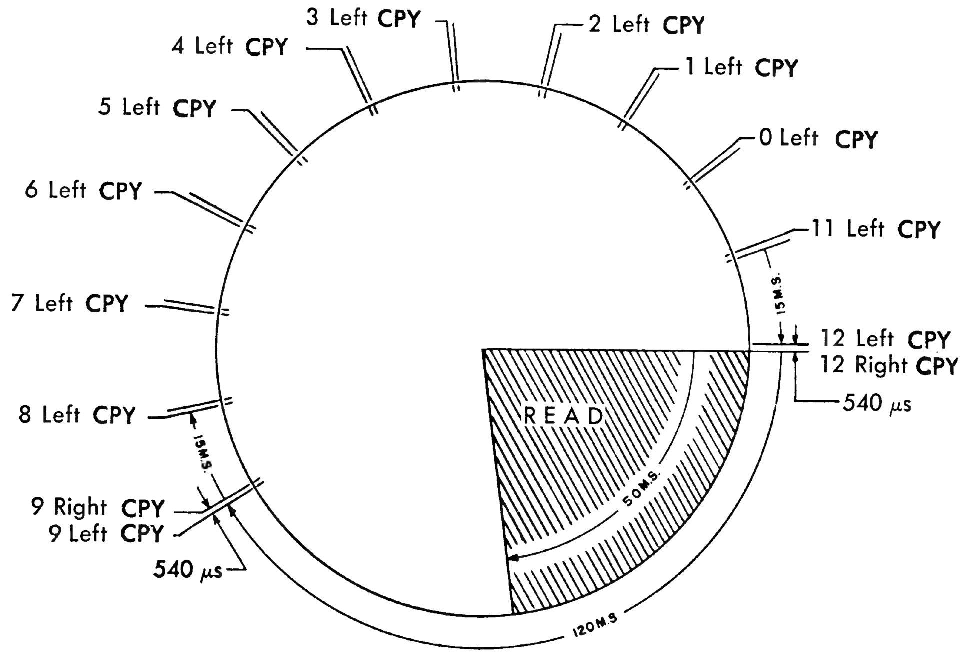


FIGURE 25

and to keep the card reader in continuous motion, it is necessary to give the RDS between 20 and 50 ms after the 12-right CPY.

If the card reader is not in motion and an RDS is given, the average elapsed time between the RDS execution and the first 9-left CPY execution will be 270 ms. However, only 50 ms are available for calculation after the RDS is given.

Timing for Model 2

Cards are read at the rate of 250 per minute. In continuous card reading, 180 ms of the card cycle of 240 ms are available for computing. The difference, 60 ms, is required for the execution of the CPY and RDS instructions and appropriate time-margins for safe synchronization of mechanical and electronic components.

The maximum safe times available for computing between executions of CPY instructions are indicated in Figure 26. As in the Model 1 card reader, 540 μs are available for computing between the left and right CPY instructions. However, there are only 8 ms available for computing between the right and left CPY instructions. The RDS must be given in the hatched portion for continuous operation of the card reader. After the 12-right CPY, it takes 12 ms for the calculator to disconnect the card reader. No input-output instructions can be executed until the card reader has disconnected. If the RDS, then, is given *t* ms after the 12-right CPY, and if *t* is less than 12, the calculator will compute for these *t* ms. Upon receiving the RDS, however, the program will be delayed until 12 ms (from the 12-right CPY) have elapsed. If the RDS is given after the 12 ms interval, the program will not be delayed. So to be able to compute for *all* of the available time between cards, *and* to keep the card reader in continuous motion, it is necessary to give the RDS between 12 and 30 ms after the 12-right CPY. If the RDS is given between 30 ms and 90 ms after the 12-right CPY, the card reader will stop and start again with a loss of only 60 ms.

If the card reader is not in motion and an RDS is given, the average elapsed time between the RDS and the 9-left CPY execution will be 110 ms. However, only 55 ms are available for computing after the RDS is given.

Manual Operation

To prepare the card reader for control by the calculator, once the control panel is in place, it is necessary only to fill the hopper with cards and hold the start key until the ready light goes on. Figure 27 shows the card path through the card reader, and indicates the relative locations of the card levers, con-

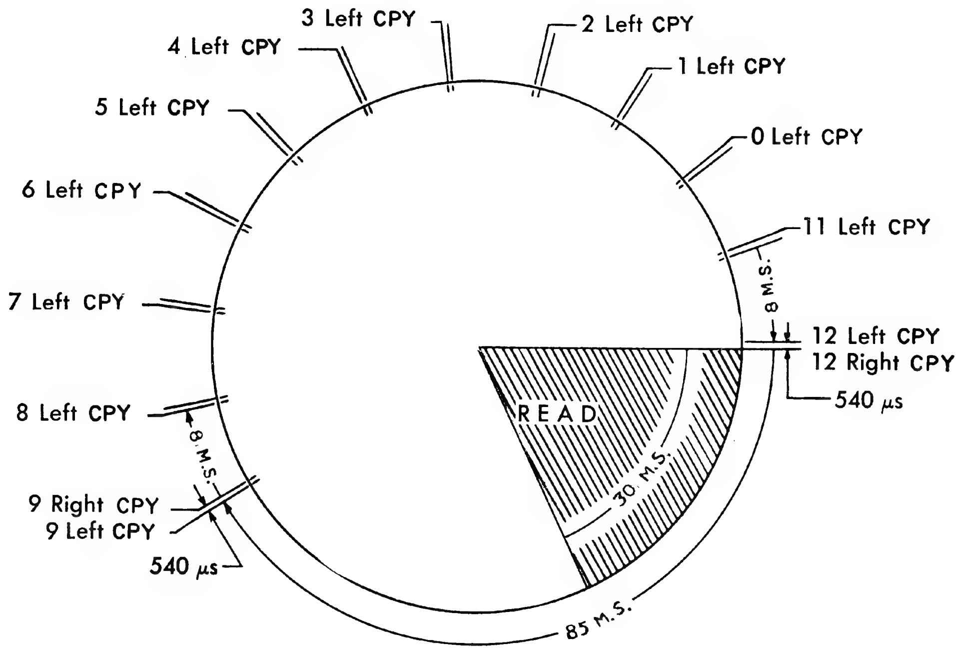


FIGURE 26

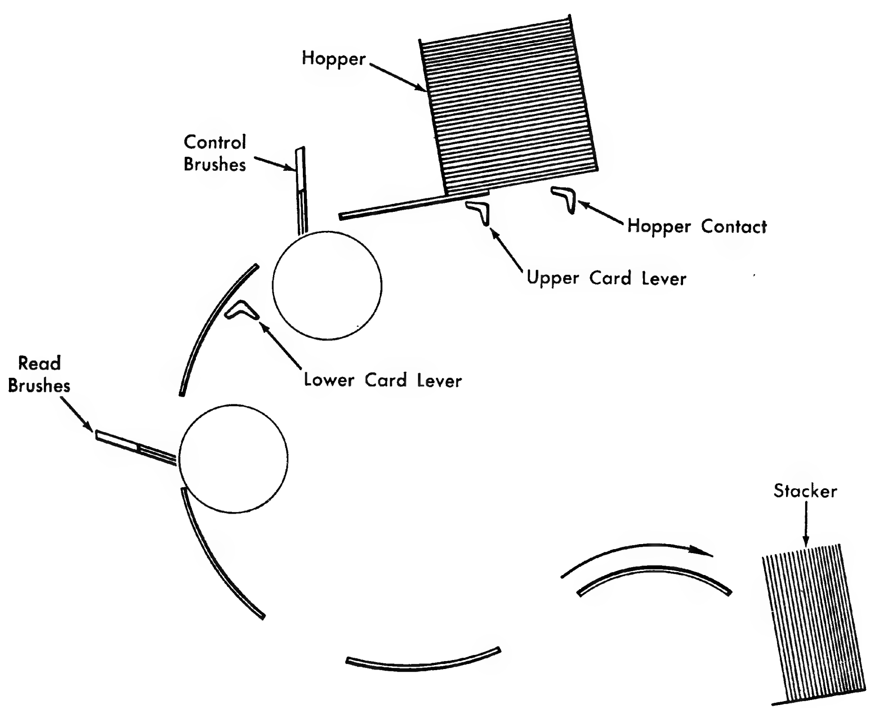


FIGURE 27

tacts, and reading brushes as cards move through the reader under control of the stored program. After the card reader has been prepared for calculator control and the ready light is on, there are two cards in the reader, and all three card contacts (upper-card lever, lower-card lever, and hopper contact) are closed.

KEYS AND LIGHTS

Start Key. Serves to run in cards initially and to turn control of the card reader over to the calculator. The key is operative only if the power is on, no fuses are blown, there is no card-feed failure, the stacker is not full, the control panel is in place, and the control panel calculator switch is wired ON.

If there is no card waiting ahead of the read brushes, press the start key to operate the card feed for one or more card cycles until the key is released or until the card enters the station just ahead of the read brushes. When the first card reaches the station ahead of these brushes, the start key causes control to be turned over to the calculator and the ready light to go on.

If there is a card waiting ahead of the read brushes, pressing the start key merely turns control over to the calculator, and the ready light is turned on.

If there are no cards in the hopper or in the card feed ahead of the read brushes, pressing the start key turns on the running light and allows the calculator to set up an end-of-file condition.

While the ready light is on, the start key cannot be used to feed cards.

Stop Key. Causes the calculator to lose control over the card reader, and turns off the ready light. If a card is being read at the time the stop key is pressed, the action is delayed, and the card reader does not stop until the end of the current card cycle. The calculator then holds up on the next CPY that refers to the card reader.

Feed Key. Permits cards to be run out of the card feed manually when the card reader is *not* under control of the calculator.

If the power is on, no fuses are blown, the stacker is not full, and the ready light is off (indicating that the calculator does not have control), pressing the feed key causes the card feed to operate for one or more card cycles until the key is released.

While the ready light is on, the feed key is inoperative. The stop key may be used to turn off the ready light in order to operate the feed key.

Ready Light. Indicates that the card reader is under control of the calculator. The ready light is turned on by the start key. It is turned off as follows:

1. By the stop key.
2. When the lower card lever is open at the end of a card cycle.
3. When the hopper contact opens at the end of a card cycle (after which it may be turned on again by means of the start key).
4. When there is a card-feed failure.
5. When a fuse is blown.
6. When the power goes off.
7. When the control panel is removed.
8. When the stacker is full.

The hopper contact opens when the hopper runs out of cards. This turns off the ready light and stops the card reader. The card reader can be started again by pressing the start key, regardless of whether more cards meanwhile were placed in the hopper.

Select Light. Goes on when the calculator gives an RDS instruction for this card reader. The light goes off when the card cycle called for by the RDS instruction has been executed.

Card-Feed Stop Light. Is on whenever there is a card-feeding failure.

Power-on Light. Indicates that the DC power is on in the card reader.

Fuse Light. Indicates a blown fuse, if the main power is still on.

CARD-FEED FAILURE

When a card-feed failure occurs, the card-feed stop light is turned on. The start key is inoperative until the following procedure is accomplished.

1. Remove all cards from the hopper . (Note that this opens the hopper contact and turns off the ready light.)
2. Run out the cards in the feed by using the feed key.
3. Press the stop key.

The last card in the stacker will not have been read.

The stop key will not reset the card-feed stop light if there are still cards in the hopper or in the feed ahead of the read brushes.

END-OF-CARDS PROCEDURE

When the last card in the hopper is fed, the hopper contact opens, the calculator stops, and the ready light is turned off.

If, at this point, there are more cards for the card reader to read, it is necessary only to reload the hopper and press the start key. The calculator will then read the cards in the hopper as if they were a continuation of the previous sequence of cards.

If, on the other hand, the card hopper is left empty when the start key is pressed, it is an indication that the end of the card file has been reached. In this case, pressing the start key will again return control to the calculator, but as the last of the remaining cards passes the read brushes, the calculator sets up and end-of-file condition; this provides a means of control by the stored program.

Timing Chart

Figure 28 is a simplified timing chart of the Type 711 Card Reader. Each machine cycle of both the model 1 and the model 2 card readers requires a certain number of milliseconds to perform a series of operations relating to reading a card. The total number of milliseconds required to complete a machine cycle is split into 20 units called cycle points. Because the card reader cycle is further divided into 360°, each of the 20 cycle points is 18° of a card reader cycle.

The card reader has an index, as shown in Figure 28, which rotates in synchronism with mechanical units in the machine. An asterisk at 330° indicates the starting time in relation to the complete cycle.

Card Cycles. Each cycle that the card feed mechanism is set in motion, a 2° to 218° impulse is emitted from the card cycles hubs.

Digit Impulses. Each cycle that a card is read under control of the calculator, the digit impulses are available at the digit impulse hubs.

Control Brushes and Reading Brushes. Control brush and read brush pulses are available at these brush hubs as cards are read at the brush stations.

Split Column Control. These pulses are available between the normal read impulse times during each feed cycle.

Pilot Selector. The pilot selector transfer relays may be energized by an 11 or 12 impulse or a digit

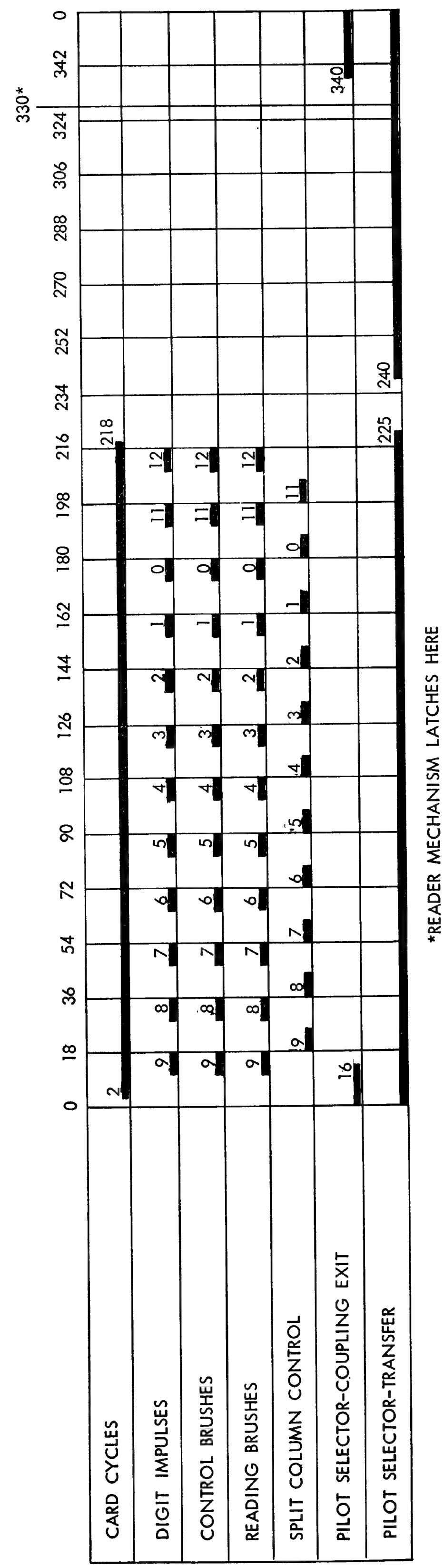


FIGURE 28

impulse (9 through 12). The selector transfers at 240° of the cycle. It remains transferred until 225° of the next card feed cycle.

For immediate transfer of the pilot selector, the immediate hubs should be impulsed. These hubs are receptive from 280° of a card feed cycle until 230° of the following card feed cycle.

Pilot Selector—Coupling Exit. Whenever an 11-12 PU or 9-12 PU hub is used to energize a pilot selector control relay, a pulse is available at the corresponding coupling exit hub the following cycle from 340° until 016°. The coupling exit impulse usually controls co-selectors. The transfer time of a co-selector can be from 252° of one card feed cycle until 225° of the following card feed cycle.

Control Panel Wiring

The card reader is a modification of the IBM Type 402. The following descriptions take advantage of similarities to the standard machine. All new functions and features are described in detail. Questions on how standard features operate can be answered by referring to the appropriate principles of operation manual.

Note that a given CPY can read only 36 columns of a given card row, because a CPY can handle no more than 36 bits. The control panel for the reader is supplied with 72 entries to the calculator, corresponding to two words in core storage or two half-rows on the card. Pulses from the reader synchronized with the passage of the card rows under the read brushes, cause first the left 36 entries and then the right 36 entries to be activated for each card row as the row passes under the read brushes. The card columns can be wired to the calculator entry hubs in any order. This section assumes that the first 72 columns of the card are used and that the entries from the calculator are used in a normal left to right order.

Figure 29 shows the hubs on the control panel of the card reader and the wiring necessary to provide for a direct transfer of information from cards in the card reader to the calculator when the proper set of instructions is provided by programming.

Hubs not described in the standard manuals or hubs with different names are described here.

Control Brushes. Equivalent to the second reading brushes of the Type 402.

Read Brushes. Equivalent to the third reading brushes of the Type 402.

Calc Entry Left and Calc Entry Right. Entry hubs for pulses entering core storage from rows of the card. Successive CPY instructions are associated alternately (by internal circuits) with first the left hubs then the right hubs, and so on. Thus, with the wiring shown in Figure 29 it is seen that successive CPY instructions will cause half-rows of the card to be read in sequence (i.e., 9-row left, 9-row right, and so on). The S hub for each of these groups accepts a pulse to determine the sign of the word being entered, while the remaining hubs numbered 1-35 correspond to the other 35 bits of the word.

On. These hubs must be connected for the card

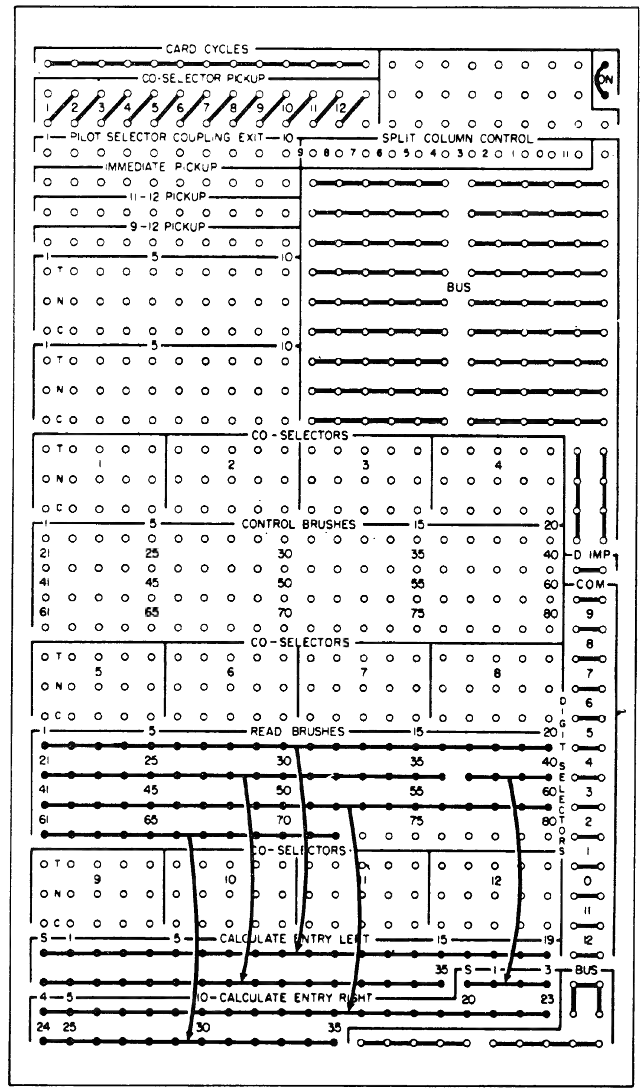


FIGURE 29

reader to operate as a component of the 704.

Conditional transfer of information is possible through the use of selectors. The pilot selectors (which work in the same way as the Type 402 pilot selectors) have three sets of pickup hubs: immediate pickup (equivalent to the *pickup* function of the immediate pickup and coupling exit of the 402); 11-12 pickup (equivalent to the X pickup of the 402); and the 9-12 pickup (digit pickup in the 402).

The pickups can be activated directly from punching in the card by wiring from control brushes to the appropriate pickup, as described in detail in the *Principles of Operation* manual for the Type 402 under the heading "Pilot Selectors." In addition, each type of pickup can be activated by an impulse from a given set of the hubs described below. In most cases, whether a given pickup will be activated by a given hub can be decided on the basis of whether the equivalent 402 pickup would be activated. Cases that cannot be decided in this way are described in detail under the appropriate heading.

By wiring the pilot selector coupling exits (similar to the coupling function of the immediate pickup and coupling exits of the 402) to appropriate co-selector pickups, one or more co-selectors can be picked up in unison with the pilot selector when the pilot selector is picked up with either the 9-12 pickup or the 11-12 pickup. In this case, the co-selector, once picked up, will hold through the next cycle in the same manner as the pilot selector.

The hubs that can activate the pickup hubs of the selectors are, as on the 402, digit selector, split column control, and card cycles. These hubs emit pulses in exactly the same manner as the Type 402.

CARD PUNCH (RECORDER) TYPE 721

THE OPERATION of punching information on a card is very similar to that of card reading. To make use of these similarities, it is necessary to understand programming for card reading before studying the following procedure.

Punching a card requires a WRS having an address of 225 (card-punch identification) to set the card-feeding mechanism of the punch in motion.

A succession of CPY instructions follows WRS. The address parts of the instructions give the locations in core storage for the words to be punched in the half-rows of the card. To punch a full card, 24 CPY instructions must be given. These CPY instructions are called, as in card reading, 9-left CPY, 9-right CPY, and so on. A separate WRS must be given for each card to be punched. Corresponding to card reading, a certain amount of computing can be carried out between WRS and the first CPY and between successive CPY instructions.

For example, binary to decimal conversion can be completed while a card is being punched. Each CPY however, must have been given by the time the corresponding half-row appears at the punching station. These time limits are specified below. If time limits are exceeded, the machine stops, and the read-write check light on the operator's console signals the error. To keep the punch running at its full speed of 100 cards per minute (and if more than one card is to be punched) give succeeding WRS instructions within a certain time interval. Otherwise, if a WRS instruction is delayed too long, the punch will stop and will not start again until the WRS is actually given. If these WRS and CPY instructions are given in succession, the calculator delays until a half-row is actually in position to be punched.

If fewer than 24 CPY instructions are given, the remaining half-rows on the card, for which there were no CPY instructions, are left blank.

Again, remember that the MQ serves as an intermediate storage during a CPY instruction. Since the word to be punched first enters the MQ before being sent to the card punch, any previous information in the MQ is destroyed.

There are no end-of-record or end-of-file conditions in punching cards.

Timing

Cards are punched at the rate of 100 cards per minute. In continuous card punching 442 ms of the card cycle of 600 ms are available for computing. The difference, 158 ms, is required for the execution of the CPY and WRS instructions and appropriate time-margins for safe synchronization of mechanical and electronic components.

The maximum safe times available for useful calculating between executions of CPY instructions are indicated in Figure 30. For example, after execution of the 9-left CPY, 540 μs are available before the 9-

right CPY must be given; and after execution of the 9-right CPY, 31 ms are available before the 8-left CPY execution. The WRS must be given in the hatched portion for continuous operation of the punch. After the 12-right CPY execution of a card, however, it takes 25 ms before the machine can execute a WRS for the next card. If the WRS, then, is given *t* ms after the 12-right CPY, and if *t* is less than 25, the calculator will compute for these *t* ms. Upon receiving the WRS, however, the program will be delayed until 25 ms (from the 12-right CPY) have elapsed. If the WRS is given after the 25 ms interval, the program will not be delayed on this account, but the punch will already have disconnected, and a delay results. Thus, to be able to compute for *all* of the available time between cards, *and* to keep the punch running at full speed, it is necessary to give the WRS at exactly 25 ms after the 12-right CPY.

If the card punch is not in motion, and a WRS is given, the average elapsed time between the WRS execution and the first 9-left CPY execution will be 400 ms. However, only 70 ms are available for calculation after the WRS execution.

If more than 24 CPY instructions are given per card cycle, the calculator will turn on the read-write check light on the operator's console, and stop.

Manual Operations

Keys and lights on the punch are similar to the corresponding controls on the card reader. Consequently, the discussion of punch controls is of a general nature. Details peculiar to the punch, however, are explicitly discussed. Note particularly that there is no end-of-file condition on the punch.

To turn control of the card punch over to the calculator (once the control panel is in place with the calculator hubs connected) the hopper is filled with cards, and the start key is held until the ready light goes on. There will now be one card in the punch; the hopper contact and the die-card-lever contact will be closed. When an appropriate program is executed by the calculator, the first card that was in the hopper will be punched with information from core storage.

The path of the cards through the punch is shown in Figure 31 along with the relative locations of the card levers, brushes, and punches. If, for any reason, one of the card contacts is not closed, the ready light

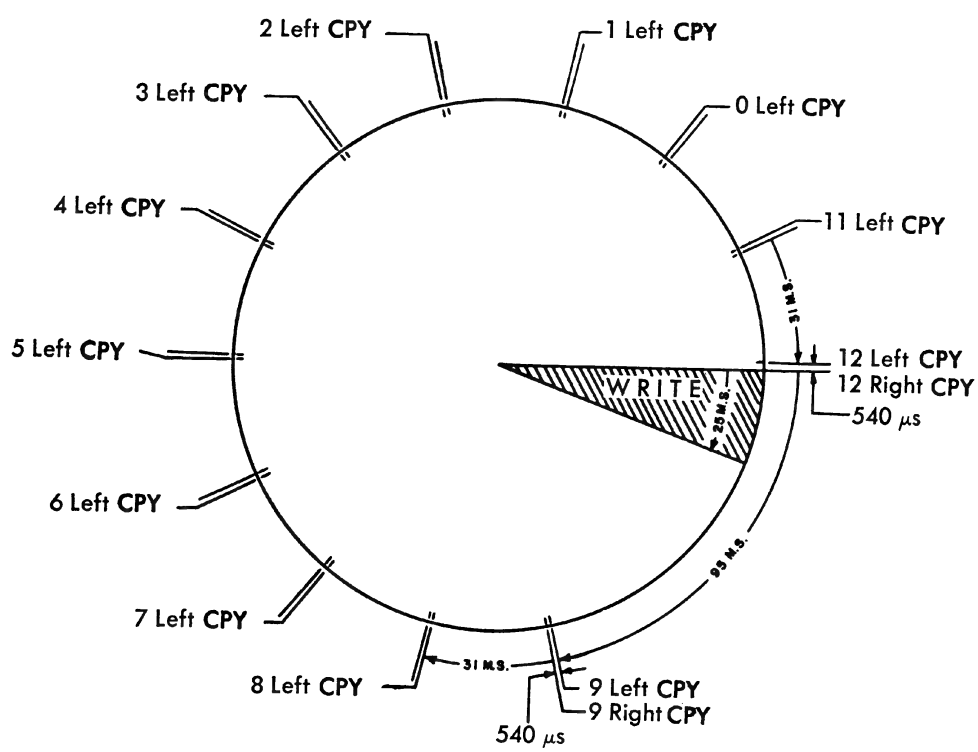


FIGURE 30

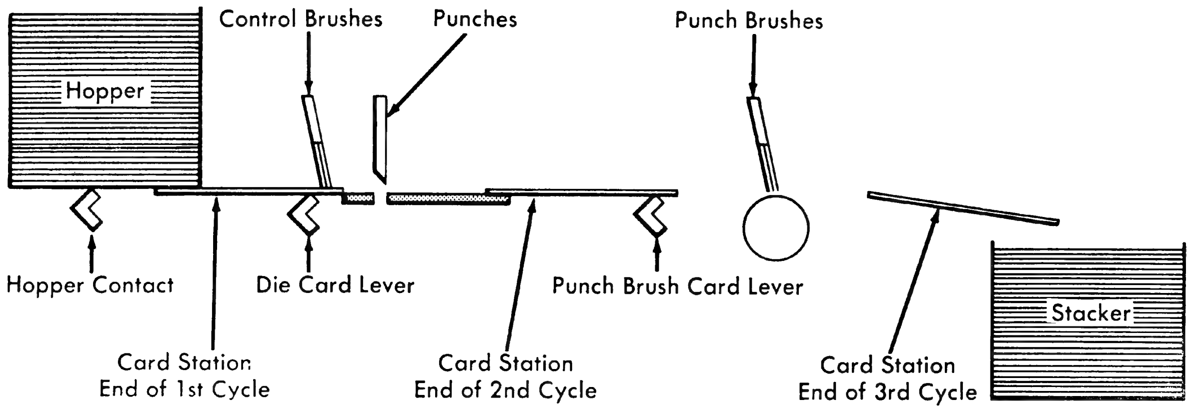


FIGURE 31

will be turned off. The ready light is also turned off by any of the following conditions in the card punch: power off, blown fuse, full stacker, control panel not inserted, gang-punch switch wired, or a double punch or blank column if the panel is so wired.

If the control panel has been so wired, the double punch or blank column in any of the columns being checked will turn on the double-punch blank-column light and turn off the ready light. To reset the double-punch blank-column light, press the stop key. To return control to the calculator, press the start key.

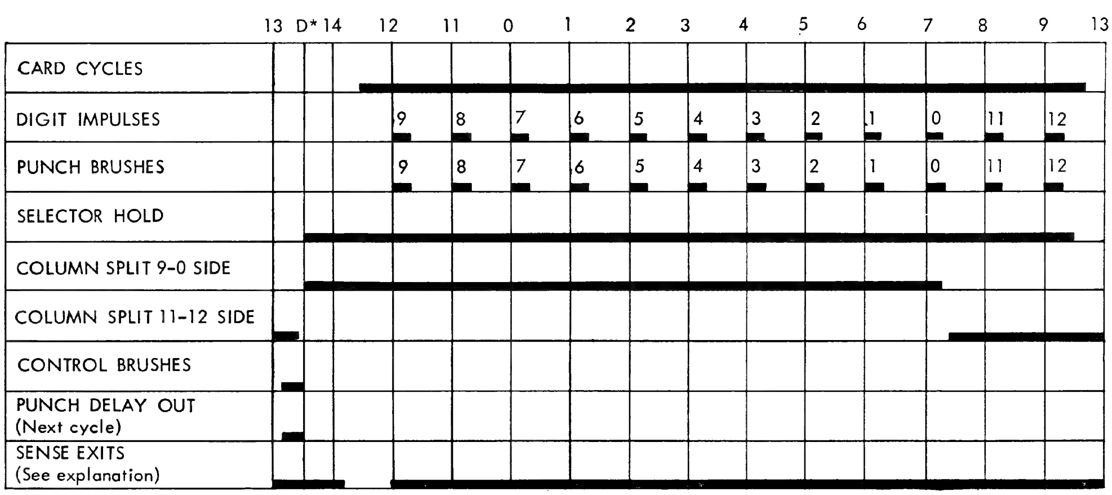
If the gang-punch (GP) switch on the control panel is wired, the ready light cannot be turned on, so the calculator has no control over the card punch. The punch can now be used as a gang punch in the usual way.

If the control panel has been wired for gang punching with the calculator switch on the control panel wired, the operation depends upon the condition of the ready light. If the ready light is on, then each card fed under calculator control will be gang punched in accordance with the panel wiring. (The feed key under this condition is inoperative.) If the ready light is off, then gang punching takes place as long as the feed key is depressed; the feed key must be held down, for as soon as it is released, the punch stops at the end of the next card cycle.

Should the card punch run out of cards, the hopper contact opens, and the ready light is turned off.

Timing Chart

Figure 32 is a simplified timing chart of the Type 721 Card Recorder. Each machine cycle of the card punch requires 600 ms to perform a series of operations relating to punching a card. The 600 ms are split into 14 cycle points. The punch cycle is further



*PUNCH MECHANISM LATCHES HERE

FIGURE 32

divided into tenths of a cycle point.

The index of the card punch begins at 13.5 or D time. Note that the index is marked off as though the cards are fed 12-edge first. It is necessary to convert 12 time to 9 time, 11 time to 8 time, and so on when thinking of the position of a card row in relation to the index.

Card Cycles. Each cycle that the feed mechanism is set in motion, a pulse is available at these hubs.

Digit Impulse. These impulses are available each cycle that the feed mechanism is set in motion.

Punch Brushes. Pulses of the duration shown are made available at the punch brush hubs while a card is being read at this station.

Control Brushes. Control punches in the 8-row of a card are sensed at 13.1 to 13.5 of a punch cycle.

Selector Hold. Selectors are held transferred until 9.5 of a punch cycle after being picked up by any punch brush pulse or digit impulse. An impulse from a control brush may also transfer a selector. This means that the selector points transfer at the end of a punch cycle and are held transferred until 9.5 of the succeeding punch cycle.

Punch Delay Out. An impulse is available one punch cycle following the sensing of a control punch (punch control brush wired to punch control in).

Column Split. The column split acts as an internally wired selector that transfers from the 9-0 side to the 11-12 side after 0 impulse time (7.4 on the machine index). The contacts return to their normal positions at 13.4 time.

Sense Exits 1 and 2. The cam that controls the sense exits is made from 14.9 until 14.2 of the following feed cycle as shown in the chart. To be certain that a PSE having a punch address is effective, the PSE should be programmed immediately following the WRS or sometime after the first CPY.

Usually an impulse from a sense exit hub controls selectors. It is usually desirable to have the selectors transferred at the beginning of the punch cycle. In general, the PSE follows the WRS. If so, the impulse available at the sense exit hub lasts until 14.2 of the punch cycle. By this time, the selector relays have established their hold circuits.

Whenever selectors are to be energized during a cycle, remember that at least six milliseconds are required to transfer the selector points.

If a PSE is programmed after a WRS and the punch is not ready, the pulse is available at the sense exit hub until 14.2 of the first calculator controlled punch cycle.

Caution note: If a PSE is programmed after a CPY and the sense exit pulse is used to control a selector, the selector transfers for the remainder of the current cycle. Because the sense exit pulse is available until 14.2 of the next punch cycle, the selector is held transferred an additional cycle.

Control Panel Wiring

The Card Punch for the Type 704 is similar to the IBM Type 521, and only new panel functions are described here.

Figure 33 shows the hubs on the card punch control panel and the wiring required for punching information directly from the calculator into a card, when the proper sets of instructions are executed by the program. The example provides that information from the calculator be punched in columns 1-72 of the card; but, as in the card reader, it is possible to punch *any* arrangement of not more than 72 columns by using appropriate wiring. The hubs involved in wiring this panel are explained below.

Punch Magnets. These hubs have the same function as the corresponding hubs on the Type 521.

Calc Exit Left and Calc Exit Right. These are exit hubs for information being transmitted from core storage to the punch magnets. In relation to programming, the word specified by the first CPY following the WRS is available at the CALC EXIT LEFT hubs. The word specified by the second CPY is available at the CALC EXIT RIGHT hubs. The left and right hubs then alternate for the following CPY instructions. Because cards are fed 9's edge first, it is evident from the wiring in Figure 33 that the half-rows of the card are punched in sequence (i.e., 9-row left, 9-row right, and so on).

CA (calculate). These hubs must be wired to enable the punch to operate as a component of the 704.

The four selectors (two standard) of 10 positions each are picked up by means of the corresponding selector pickups, 1-4. Activation of the pickups directly from the control brushes will be discussed later, using gang punching for illustration purposes. The selectors can also be picked up by electrical im-

pulses available at the two sense output hubs in the upper-right section of the control panel.

Sense Output. Wiring a sense output hub to a selector pickup allows a programmed PSE with the proper address to transfer a selector. A PSE given at any time after the first CPY of a given cycle, and at least 32 ms before the first CPY of the *next* cycle, causes the selector to be transferred shortly after the instruction is given (between 15 and 30 ms later). The selector then stays transferred until 20 ms after the last CPY of that next cycle. In normal usage, the PSE is given soon after the WRS that initiates the cycle in which the selector is to be picked up. When all rows of a card are not being punched after a WRS any gang punching or emission of digits directed

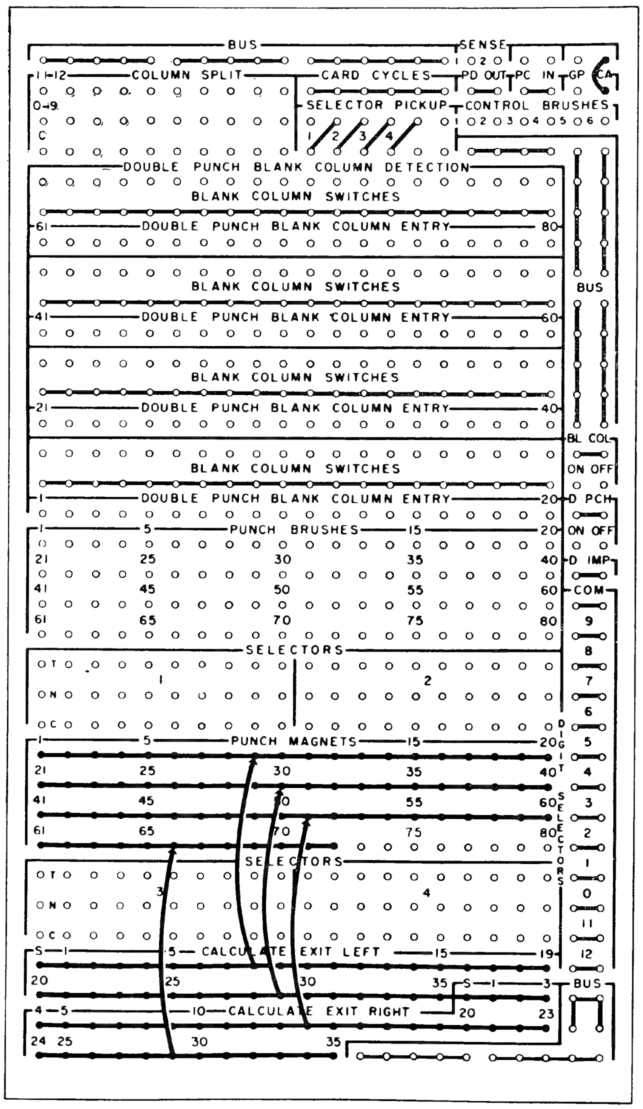


FIGURE 33

through selectors, which in turn are controlled by sense exits, requires special precautions to insure that the selector is transferred *only* during the cycles desired. The address of sense output hub number 1 is 225, while number 2 is 226.

All other hubs on the panel operate in the same manner as on the standard Type 521.

The card punch can be used for gang punching cards under calculator control as well as independently of it. Figure 34 shows the wiring necessary for gang punching with interspersed master cards under calculator control. (For this example the information to be gang punched is assumed to be in columns 75-80 of the card.)

The control panel wiring as shown in Figure 34 is:

1. Hubs S-35 of the calculator exit left followed by hubs S-35 of the calculator exit right are wired to hubs 1-72 of the punch magnets.
2. Columns 75-80 of the punch brushes are wired to five consecutive common hubs of selector 2. The corresponding normal hubs are wired to columns 75-80 of the punch magnets.
3. The hub for control brush 1 (positioned at the column in which the control punch will be) is wired to selector pickup 2.
4. The CA hubs are connected.

To gang punch cards independently of the calculator, the GP (gang punch) hubs should be connected instead of the CA hubs.

If the program should call for the card punch to operate with the GP hubs wired, the calculator will stop, because no control by the program is possible with these hubs wired.

Offset gang punching can be done in the normal way by wiring the appropriate control brush to the PC (punch control) hub and wiring the PD (punch delay) hub to the selector pickup. To prevent punching in the master cards when offset gang punching, two selectors are necessary: one wired as above to provide for offsetting, the other wired to prevent punching the master card.

PRINTER, TYPE 716

THE PRINTER, a modification of the printing unit on the IBM Type 407 Accounting Machine, is equipped with 120 rotary type-wheels. Each wheel has 48 characters, including numerals, alphabetic symbols,

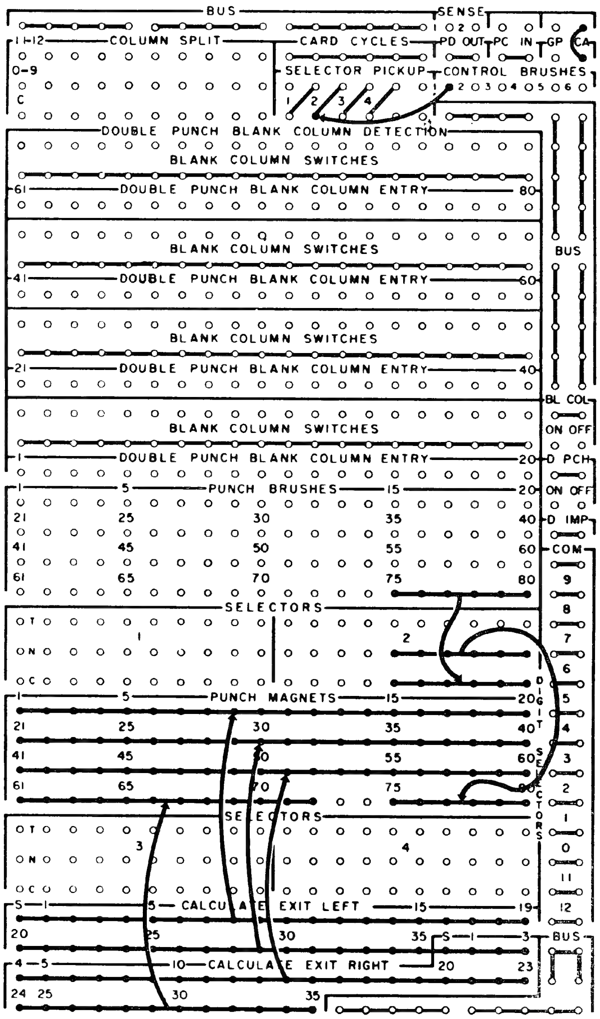


FIGURE 34

and special characters. Use of the proper program enables the machine to print decimal numbers, binary numbers, or numbers to any other base. Titles and headings are also possible, because alphabetic characters and special symbols are provided on the type-wheels.

As in the standard Type 407, the type-wheels are positioned for printing by electrical pulses timed according to the print cycle itself. Remember that if a print-wheel receives an electrical impulse during that part of the print cycle designated as 9-time, the print-wheel is positioned to print a 9. Also, if the print-wheel receives an impulse at 1-time *and* an impulse at 12-time, the machine interprets this as the letter *A* (according to the standard IBM code) and positions the type-wheel to print *A*. It is important to understand this timing principle of accounting machines. Refer to the Type 407 Accounting Machine Principles of Operation Manual.

The simple example below shows how a series of nines might be printed in 72 positions of a line. Example: A WRS with an address of 241 (printer identification) is programmed, causing the printer to start a print cycle. The print cycle, as it progresses, goes through points in the cycle known as 9-time, 8-time, 7-time, and so on to 11-time and 12-time. These times are analogous to the times designated for the standard Type 407, which operates in conjunction with a card-reading mechanism.

In the standard Type 407, the above times coincide exactly with the time the 9-row is under the reading brushes, and so on. As soon as the 704 printer reaches 9-time in its cycle, the program must furnish a CPY to read a word from a core storage location. By using control panel wiring this full word is directed to 36 type-wheels (one for each bit of information) of the printer. A second CPY then follows causing another word in core storage to be directed to 36 other type-wheels. These two CPY instructions are given close enough together (with respect to time) for the printer to still be essentially at 9-time of its cycle. A binary digit of 1 in the full word causes an electrical pulse to enter the printer and to impulse the associated type-wheel. This impulse causes the type-wheel to be positioned for printing a 9, because the impulse arrived at 9-time of the print cycle.

If now we assume that both full words described with the above two CPY instructions consist of a negative sign and 35 binary ones, the result will be 72 nines printing across the page. This assumes that any subsequent CPY instructions corresponding to 8-time, 7-time, and so on do not cause an additional impulse to a type-wheel and thus cause the wheel to be positioned differently. A positive sign or a binary zero does not start a pulse to the type-wheels.

The general procedure in printing a line is to set up in core storage a card image similar in nature to the card image produced when a card is read by the card reader. A WRS then is followed by 12 *pairs* of CPY instructions; these cause the card image to send impulses to the type-wheels. The first pair of CPY instructions causes 9-time impulses to be sent to the type-wheels as explained in the example above. The second pair causes 8-time impulses to be sent to the

type-wheels. This procedure continues until the 12 pairs of instructions are executed in accordance with the 12 distinct times of the print cycle.

Subsequent WRS and CPY instructions produce a new print cycle and a new printed line. The word impulses, brought about by execution of the *first* CPY of a pair, are available at the CALC EXIT LEFT hubs on the printer control panel and may be directed to the selected type wheels by wiring. The impulses produced by the *second* CPY of a pair are available at the CALC EXIT RIGHT hubs. The 24 CPY instructions are called 9-left CPY, 9-right CPY, 8-left CPY, 8-right CPY, and so on to the 12-right CPY.

As in the card reader and card punch, it is possible to do useful calculating between the actual printing instructions. For example, the time required for the print cycle to start, and move to 9-time of its cycle, may be used for other calculations. Once the cycle has reached 9-time, however, the program must provide, in succession, the pair of CPY instructions for impulsing the type-wheels. Useful calculating can also be performed between pairs of CPY instructions and even between individual instructions of a pair. As before, there are definite time limits to be observed. These are precisely specified below.

If a CPY arrives too late in the cycle, the calculator stops and the read-write check light indicates the error. For the calculator to print at its full rate of 150 lines per minute, the WRS instructions for each print cycle must be given in the interval of time explained below. If WRS instructions do not follow each other within this time limit, the printer stops and will start again only on recepit of the next WRS. If we do not want to do computing between these input-output instructions, we can program these instructions in immediate sequence. Under these conditions, the calculator automatically waits until the printer reaches the proper point of its cycle before executing the instructions. Also, it is not always necessary to give a full set of 24 CPY instructions for each line of print. If a full set is not given, the action is similar to the card reader; namely, the print cycle continues without any more impulses to the type-wheels, and a following WRS starts a new cycle.

Again, the MQ is used as an intermediate storage for a word passing from core storage to the printer; so the execution of any CPY destroys information previously standing in the MQ.

Printing with Checking

The previous paragraphs give the procedure for printing *without* checking. Checking is possible because the printer is capable not only of receiving print pulses from the calculator to set up the type-wheels for printing, but also of sending to the calculator "echo pulses" generated by the type-wheels depending on what character the wheels are in position to print. Printing with checking requires a somewhat more complicated program, but it can be done without reducing printing speed. The timing for this combination is *roughly* as follows:

The first half of the print cycle is used to position the type-wheels by using words in core storage. The second half is used for reading the "echo pulses" generated by the type-wheels and for placing them in core storage for verification via a programmed check.

When we wish to check, the echo pulses are read in such a way as to form a card image when received by the calculator. Thus, the calculator can both write the original card image for printing and read a corresponding card image, at a later time in the same print cycle, from the echo pulses. If the two card images do not agree exactly, as determined by a suitable program, then an error has occurred. Only numerical information can be checked in this way.

Program the printing with checking as follows. First give an RDS (note this difference), with the address of the printer followed by *46* CPY instructions in a specified sequence. Twenty-four of these CPY instructions refer to printing, and cause words to be sent from core storage to the printer. The other *22* CPY instructions refer to checking. They require words to be read from the printer into core storage. During part of the print cycle, the two kinds of CPY instructions must alternate in pairs. Note, then, that the RDS causes *both* writing information from storage to printer, *and* reading the echo impulses into storage.

Exact sequence of the 46 CPY instructions for printing with checking is described below. There are two sets of codes for plus and minus signs, as follows: with one set, used for printing *without* checking, 12 is the code for plus, and 11 for minus; with the other set, used for printing *with* checking, the combination of 8 and 3 is the code for plus, and 8 and 4 the code for minus.

The first 18 CPY instructions are to supply impulses to the printer from the left and right halves of rows

9 through 1 of the card image. The 19th and 20th CPY instructions are for storing the echo impulses received from the minus sign (code 8, 4). The 21st and 22nd CPY instructions send impulses to the printer from the zero row of the card. The 23rd and 24th CPY instructions store echo impulses received from the plus sign (code 8, 3). The 25th and 26th send the 11-row of the card image to the printer. The 27th and 28th are for checking the 9-row. The 29th and 30th send the 12-row of the card image to the printer. Finally the 31st through 46th CPY instructions form the check images of the 8-row through the 1-row. The fact that no checking is provided for the 0, 11, and 12 rows explains the two separate codes for plus-and-minus signs. The exact sequence of instructions and the allowable time between them are fully explained below.

Through use of selectors or column splits on the printer control panel, core storage can activate more than 72 type-wheels. For example, seven 10-digit numbers with signs can be printed. Additional characters can also be printed by impulses emitted on the control panel. On the other hand, the control panel can be wired so that up to 120 characters originating from core storage can be printed on each line at the rate of 75 lines per minute, two cycles being required for each line of printing.

The printer has an IBM tape-controlled carriage, details of which are described in the section on tape-to-printer operation. Functions of the carriage (such as changing or suppressing line spacing, selecting the channel on the punched tape to control skipping, or sensing sheet overflow) may be controlled through sense output or input hubs on the control panel; these hubs are activated by appropriate PSE instructions in the stored program.

Timing without Echo Checking

Information is printed at the rate of 150 lines per minute. In continuous printing, 322 ms of the print cycle of 400 ms are available for computing. The difference, 78 ms, is required for the execution of the CPY and WRS instructions and for appropriate time-margins for safe synchronization of mechanical and electronic components.

Maximum safe times available for useful calculating between executions of CPY instructions are indicated in Figure 35. For example, after execution of the 9-left CPY, 540 μs are available before the 9-right CPY must be given; and after execution of the 9-right CPY, 13 ms are available before the 8-left CPY. The WRS must be given in the hatched portion for con-

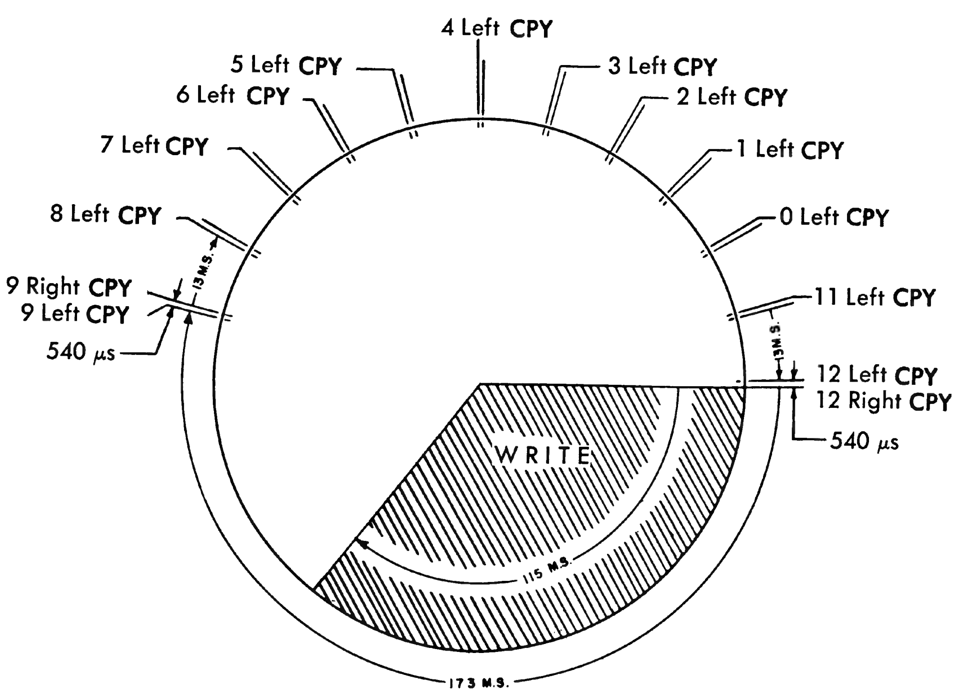


FIGURE 35

tinuous operation of the printer. After the 12-right CPY execution of a print cycle, however, it takes 16 ms before the machine can execute a WRS for the next cycle. If the WRS, then, is given *t* ms after the 12-right CPY, and if *t* is less than 16, the calculator computes for these *t* ms. Upon receiving the WRS, however, the program is delayed until 16 ms (from the 12-right CPY) have elapsed. If the WRS is given after the 16 ms interval, the program is not delayed. So to compute for *all* of the available time between print cycles, *and* to keep the printer running at full speed, it is necessary to give the WRS between 16 and 115 ms after the 12-right CPY.

If the printer is not in motion, and if a printer WRS is given, the average elapsed time between the WRS execution and the first 9-left CPY execution is 280 ms. However, only 58 ms are available for calculation after the WRS execution.

Timing with Echo Checking

In continuous printing with checking, 313 ms are available for computing. Appropriate times are given in Figure 36.

The RDS must be given in the hatched portion for continuous operation of the printer after the 1-right echo CPY execution of a print cycle; however, it takes 12 ms before the machine can execute an RDS for the next cycle. If the RDS, then, is given *t* ms after the 1-right echo CPY, and if *t* is less than 12, the machine can compute for these *t* ms. Upon receiving an RDS, however, the program will be delayed until 12 ms (from the 1-right echo CPY) have elapsed. If the RDS is given after the 12 ms interval, the program will not be delayed on this account; but the printer will already have disconnected, resulting in a delay. So to compute for *all* of the available time between print cycles, *and* to keep the printer running at full speed, give the RDS at exactly 12 ms after the 1-right echo CPY.

If the printer is not in motion and if a printer RDS is given, the average elapsed time between the RDS and the first 9-left CPY execution is 280 ms. However, only 58 ms are available for calculation after the RDS execution.

Manual Operation

Keys and lights on the printer are similar to those on the card reader, with the following exceptions:

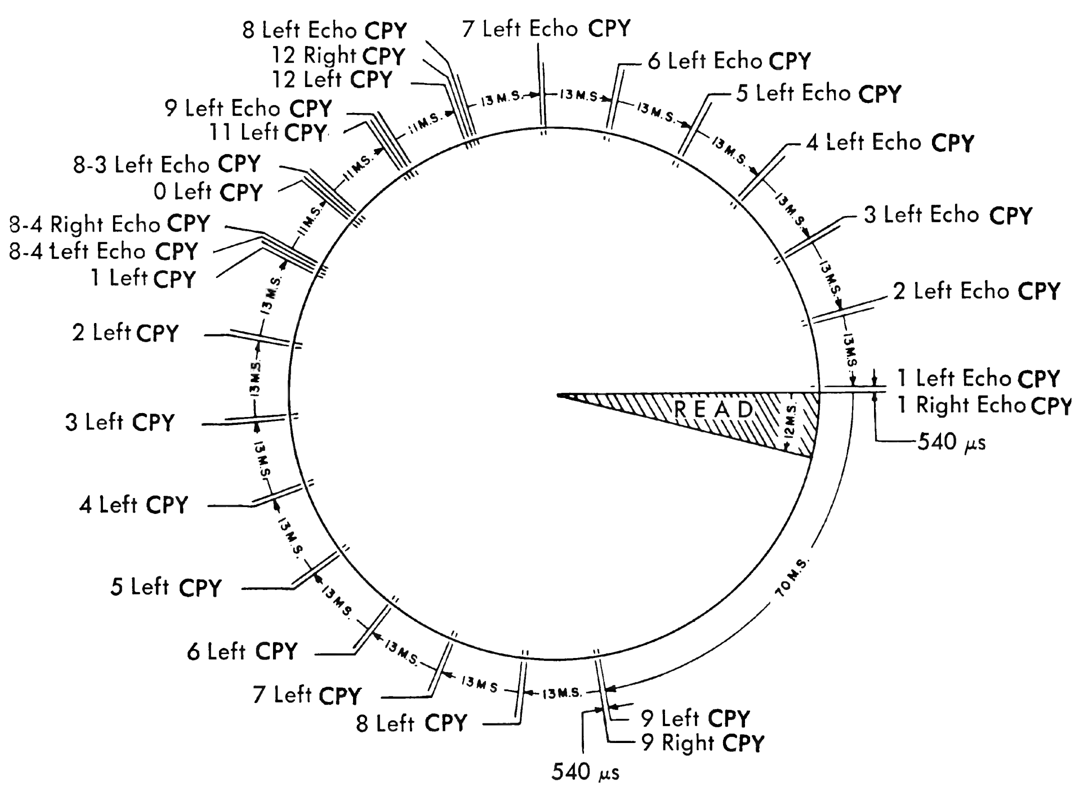


FIGURE 36

1. The feed key is replaced by a print-cycle key.
2. The card-feed stop light is replaced by a form-stop light.
3. The stop-before-printing, test, and form-stop switches are added.

The following is a general discussion of printer controls with special emphasis on the differing features.

To prepare the printer for control by the calculator—once the control panel is in place—it is necessary only to hold the start key until the ready light goes on. The ready light is turned off by any of the following conditions: test switch on; a form stop, indicated by the form-stop light, when the form-stop switch is on; depressed stop key on the carriage; depressed stop key on the printer; power off; blown fuse. (The test switch is discussed below.) If the form-stop switch is on, the form-stop light goes on when the printer runs out of paper. Other carriage controls are similar to controls on the Type 407 carriage.

As in the card reader and card punch, the printer stop key gives the operator a means of holding up printing whenever he so desires. The carriage stop key has an equivalent effect.

Turning the test switch on causes the ready light to go off.

The print cycle key starts a print cycle only under the following conditions:

1. When the ready light is off (as when the test switch is on).
2. When the ready light is on, the stop-before-printing switch is on, and the program supplies an RDS or WRS instruction for the printer.

With the test switch *on*, the ready light is off. Depressing the print cycles key causes the printer to go through print cycles until the key is released. To switch control back to the calculator, turn off the test switch and press the printer start key.

With the stop-before-printing switch on, the printer, after being selected by an RDS or WRS instruction in the program, is held up at the first CPY instruction. Depressing the print cycles key causes the printer to print one line and the calculator to proceed with the program until the next group of output instructions for the printer is ready to be executed.

Timing Chart

Figure 37 is a simplified timing chart of the Type 716 Alphabetic Printer. Each machine cycle of the

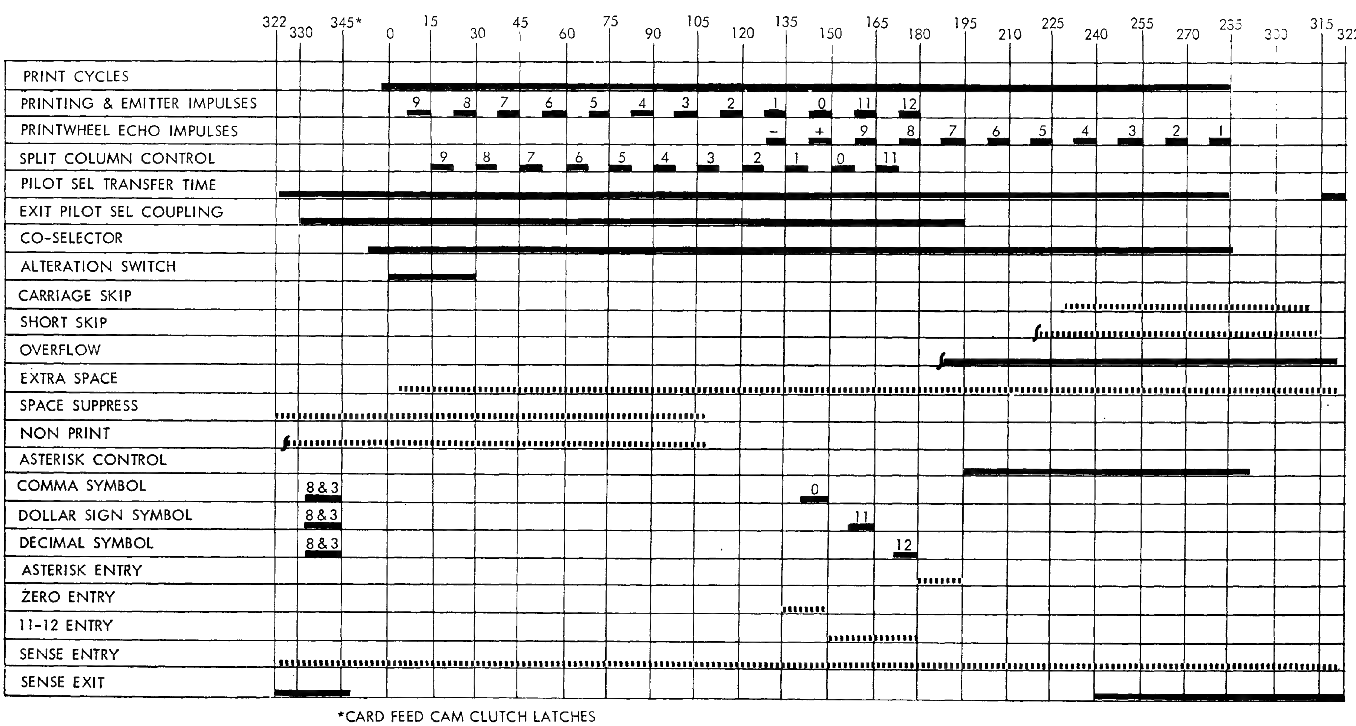


FIGURE 37

printer requires 400 ms to perform a series of operations relating to printing a line of information. The 400 ms are divided into 24 cycle points. Thus each cycle point is 15° of a printer cycle. The starting point of the printer index is 345°.

Carriage Skip. Because skipping is an afterprint operation, the carriage skip hubs are not receptive until any before-print spacing operation has been initiated. To be effective, an impulse to the carriage skip hubs must occur between 230° and 310° of a print cycle.

Space Suppress. An impulse directed into these hubs between 320° and 110° of a print cycle suppresses before-printing spacing.

Extra Space. An impulse directed into this hub after 5° and before 320° of a print cycle provides an extra-space-after-printing operation according to the printer control panel wiring.

Overflow. An impulse is available from the overflow hubs during the before-print-space operation that precedes printing of the last line on the form. If the overflow operation is not delayed, the impulse lasts until 320° of the current print cycle. Otherwise the pulse is available until 320° of the print cycle during which the overflow operation starts.

Short Skip. An impulse directed to a short skip hub before 300° of a print cycle during which a skipping operation is starting or taking place releases the interlock normally causing the printer to lose at least one machine cycle for each skipping operation.

Non-Print. An impulse directed to this hub before 110° of a print cycle prevents printing, spacing before printing, and ribbon spacing.

Split Column Control Emitter. These hubs emit impulses between normal digit impulses, i.e., the 9 hub emits an impulse between the normal 9 and 8 times of the master CB pulses.

Alterations. When the corresponding alteration switch on the switch panel is on, the alteration hub emits a 0° to 30° impulse each print cycle.

Print Cycles. These hubs emit pulses during each print cycle from 355° to 285°.

Co-Selector Pickup. A co-selector is held transferred the remainder of a print cycle if it is pulsed sometime after 350°.

Pilot Selector. A pilot selector control relay can be energized any time between 5° and 310° of a print cycle. The points of the pilot selector relay are held transferred from 315° of a print cycle until 286° of the following print cycle.

Coupling Exit. A pulse is available at a pilot selector coupling exit from 330° of the print cycle that a pilot selector control relay is energized until 195° of the following print cycle.

Sense Entry. The sense entry hub is receptive at all times.

Sense Exit. Usually the sense exit hubs are conditioned so that pulses can be emitted from these hubs between 240° and 350° of a print cycle. The sense exit pulse is available only until 340° of a cycle during which a carriage skip operation is started. The PSE can be programmed before 240° but the pulse is not emitted until 240°. Ordinarily a PSE follows immediately after the WRS or RDS.

Extra-space, space-suppress, and non-print operations may be controlled directly from sense exits.

716 Timing

Each machine cycle of the printer requires 400 ms to print a line of information. These 400 ms are divided into 24 cycle points of 15° per point. The dividing line between machine cycles is 345° when the card feed cam clutches are latched up. These clutches control all emitter impulses except the three special symbol emitters (comma, dollar sign and decimal). These three symbols are controlled by cams termed "continuously running." After a printing operation, the card feed cams have a forced drop-out at 345° while the continuously running cams are permitted to coast to a stop. The continuously running cams will drop out between 50° and 250° on the printer index.

If a print cycle is initiated when the printer is not in motion, synchronization of the 704 and 716 may take up to 400 ms and averages 280 ms. The actual selection of the printer will occur between 322° and 345°. When the print cycle is initiated after another print cycle, selection comes before 295°, no synchronization is needed, and the actual selection of the printer will occur between 300° and 345°.

Note: Care must be taken if the special symbol emitters are used to control printer operations. On a printer reselect, these emitters are operative before the card feed cam clutches are latched up. This can cause a function to occur at point in the printer

cycle when the select follows a non-print cycle, and occur at a different point when the select follows a print cycle.

Normal spacing in the print cycle occurs between 120° and 135°. If this spacing causes a punch in channel 12 of the control tape to pass the control tape reading brush, an overflow pulse is emitted starting at 155° and extending through 320°. This pulse may be used at the end of the print cycle, or, if the printer is kept selected, immediately following the print select instruction. *Note:* If the printer motion is halted, the pulse will not be available when the printer is reselected.

The overflow hubs should be wired to carriage skip 1 hub if the overflow is to be controlled only by the 716, or to the sense entry hub if the overflow is to be controlled by a sense instruction.

If the sense entry is used with the overflow hub, it is recommended that the PSE 240 be given at the end of the current print cycle, rather than after the next select instruction.

Printer Disconnect

If 24 CPY instructions accompany a WRS printer instruction, the printer disconnects between 180° and 195°. If fewer than 24 CPY instructions are given, the printer disconnects during any 15° time that two CPY instructions are not given. If 46 CPY instructions accompany a RDS printer instruction, the printer disconnects between 285° and 300°. If fewer than 46 CPY instructions are given, *two* CPY instructions must be given during any 15° time to prevent printer disconnect, except during 120° to 180° when *four* CPY instructions must be given during each 15° time to prevent printer disconnect.

Control Panel Wiring

The printer is used to print information contained in core storage. Note that this printed material can be any combination of several characters. Some characters that can be printed by impulses from core storage are decimal digits, letters of the alphabet, punctuation marks, and dollar signs. Table V gives the IBM code for printing these characters. For example, a dollar sign ($) may be printed by arranging impulses (from core storage) to arrive at a print wheel at 11-time, 8-time and 3-time of the print cycle.

Digit	No (N) Zone	12 (Y) Zone	11 (X) Zone	0 Zone
No Digit	*	+	—	0
1	1	A	J	/
2	2	B	K	S
3	3	C	L	T
4	4	D	M	U
5	5	E	N	V
6	6	F	O	W
7	7	G	P	X
8	8	H	Q	Y
9	9	I	R	Z
8-3	+	.	$	,
8-4	—	⌑	*	%

TABLE V

Figure 38 shows the control panel wiring for the printer, with the wiring for an example to be explained later. The hubs used for this example will now be explained in general terms.

Calc Exit Left and Calc Exit Right. These hubs are exits for words being sent from core storage to the printer by CPY instructions. These two sets of hubs alternate with the CPY instructions in the same way as similar hubs on the card punch.

Print Entry. These are entry hubs for impulses to the individual type-wheels from core storage.

Print Echo Exit. These hubs are exits for the echo pulses generated by the type-wheel according to the character they are positioned to print.

Calc Echo Entry Left and Calc Echo Entry Right. These are hubs that can accept the echo impulses generated by the type-wheels. The CPY instructions in the stored program then direct these impulses to core storage locations in preparation for a programmed check. The left and right hubs alternate as in the card reader.

PR, On. These hubs must be connected if the printer is to be used as a component of the 704.

ZC. If these hubs are wired together, the zero print control function behaves exactly as in the 407. If these hubs are not wired, the type-wheels print *only* if impulsed through the print entry hubs.

Printing Control

As an example, the control wiring in Figure 38 provides for printing any digit (including zero),

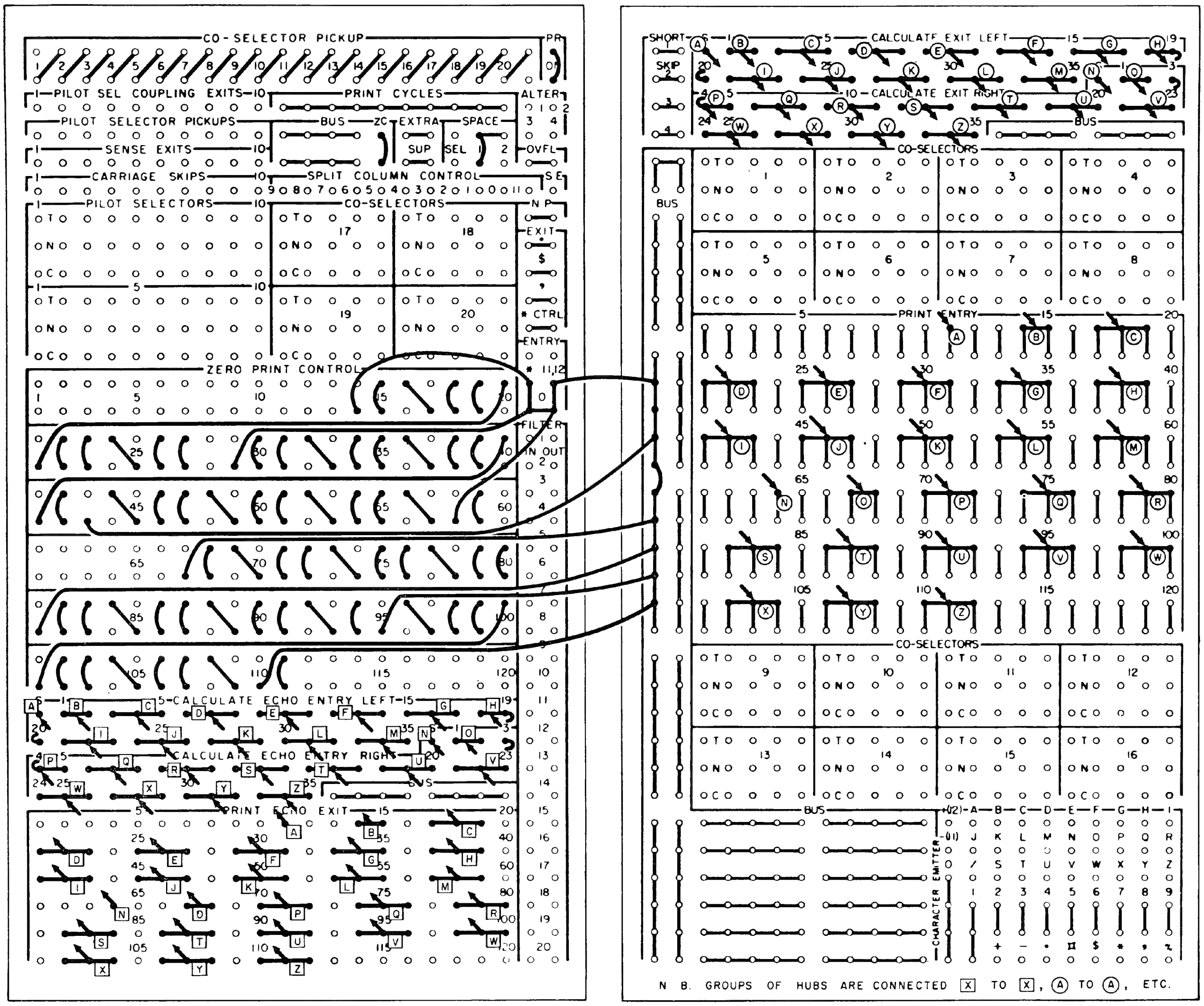


FIGURE 38

letter, or special character that has been coded in the card image being copied. The wiring also provides for echo checking of the digits 9 through 1 in all positions except 11 and 64. In positions 11 and 64, provision is made for checking the special codes corresponding to the plus and minus signs (8-3 and 8-4). Although this wiring is valid for printing any character, assume that binary numbers will be printed here.

In the specific example, characters being printed are separated into groups of three each to help translation from the binary to the octal system. Any other arrangement could be made.

The control panel wiring shown in Figure 38 is as follows:

1. The calculator exit hubs (two sets of hubs labeled S, 1-35; the left half-row and the right half-row, respectively) are wired to the print entry hubs in this order:

	CALC EXIT	PRINT ENTRY
Left half-row	S	11
	1, 2	14, 15
	3, 4, 5	17, 18, 19
	6, 7, 8	21, 22, 23
	.	.
	.	.
	.	.
	33, 34, 35	57, 58, 59

Right half-row	S	64
	1, 2	67, 68
	3, 4, 5	70, 71, 72
	.	.
	.	.
	.	.
	33, 34, 35	110, 111, 112

2. The wiring from the calculator echo entry hubs (two sets of hubs labeled in the same way as the calculator exit hubs) to the print echo exit is the same as that described in paragraph 1 above, if the words "calculator echo entry" and "print echo exit" are substituted for "calculator exit" and "print entry," respectively. Use of echo pulses is explained below.

3. The ZC hubs are wired. This allows the zero print control to operate in the same way as on the 407. Note that *all* of the zero print control wiring, described in the next paragraph, could be eliminated if the ZC hubs were not wired. The zero print control is brought into play here only to illustrate appropriate wiring.

4. The pairs of zero print control hubs (described below) are numbered to correspond to the print entry hubs. All pairs of zero print control hubs corresponding to the print entry positions enumerated in paragraph 1 are connected (i.e., the upper hub at a position is wired to the lower hub at the same position) *except* the pairs that are at the first of a subgroup (the hubs corresponding to positions 11, 14, 17, 21, . . . 57, 64, 67, 70, . . . 110). The lower hubs at positions 14 and 67 will be wired to the zero entry. A *further exception* to the system described above occurs because the zero print control hubs should not be wired in groups of greater than 10 pairs of hubs. To separate the zero print control wiring into independent groups of appropriate size, the lower hubs of some positions are wired to zero entry. The remainder of the zero print control hubs at the first of a group are connected diagonally from the lower hub of the pair to the upper hub immediately at the left (corresponding to a blank position in the printing).

5. The PR and ON hubs are coupled.

6. The space hub is wired to 1 to provide single spacing between lines of print.

To check the printing of a number and its sign, wire the print echo exits, corresponding to the print wheels that printed the number, to the calculator echo entries corresponding to the calculator exits from which the information was originally taken. It is then possible, by programming, to read these impulses into core storage and to perform a programmed check on the original information.

In general the program for this checking relies upon the fact that the echo pulses occur in a given order: 8-4, 8-3, 9, 8, 7, 6, 5, 4, 3, 2, 1. Each print wheel emits an echo pulse timed to indicate the sector within which it was set up to print. Since no provision is made for checking the zones within these sectors, the checking is restricted to numerical printing. For example, at 8-echo time, the print echo exits, corresponding to the print wheels set up to print in the 8 sector, emit a pulse. The program copies these pulses into core storage in the form of a binary word to be compared with the word in the card image corresponding to the 8-row.

Zero Print Control. With the zero print control hubs on its control panel, the 704 printer can provide any one of a number of responses to zeros or any symbol not having a digit pulse. Each pair of zero print control hubs corresponds to a print entry position; the manner in which the pair of hubs is wired to its neighbors controls the printer's response. Zero print control functions only during zone (0, 11, 12) time and *N* (no-zone) time and can have no effect upon the printing in a position that has received a digit impulse (1 through 9) during a given print cycle. Thus, the only special characters that can be controlled are those consisting only of zone pulses (the zero, check-protecting asterisk, plus sign, and minus sign) or those emitted from special hubs on the control panel (the dollar sign, period, and comma). The specially emitted symbols provide for setting the print wheel to the proper sector without use of the usual combination digit punching (as an 8 and a 3 in the case of the dollar sign, period and decimal point, and comma). The dollar sign, period, and comma can be printed with the usual combination punching, of course, but in this case the symbol cannot be controlled by means of zero print control.

Note that zero print control hubs cannot operate correctly when used in groups of more than ten at a time. Groups larger than this should be split and wired independently.

Examples and applications of zero print control are given under the heading "Zero, Comma, Decimal and Dollar Symbol Control" in the *Type 407 Prin-*

ciples of Operation manual, Form 22-5765.

Filter in-out. These hubs permit the passage of an impulse in only one direction—into the IN hub and out of the OUT hub. Do not wire the OUT hub of one filter to the IN hub of another filter, either directly or indirectly.

The printer has ten pilot selectors, each of which is picked up by an associated one of the pilot selector pickups. (These pickups are similar to the IMMEDIATE-PU hubs on the Type 407.) In addition, there are 20 co-selectors, each with two identical co-selector pickups. Pick up the co-selectors in unison with given pilot selectors by wiring the appropriate pilot selector coupling exits to the co-selector pickups; or pick up the co-selector independently by direct wiring from other hubs. The pilot selectors and co-selectors are similar, *in action,* to the pilot selectors and co-selectors of the card reader. The following hubs provide pulses to activate the selectors through their pickups.

Alteration Switches. These switches function the same as the 407 alteration switches, by emitting a pulse every machine cycle when the corresponding toggle switch on the printer is turned on. These pulses are used to pick up either pilot selectors or co-selectors.

Split-Column Control. These hubs perform the same function as the 407 split column control. The numbers on either side of a given hub of the split column control refer to the corresponding print times. A selector pickup wired from a given one of these hubs (the hub between numbers 8 and 7) causes the selector to be transferred between the corresponding print times. (The selector would be transferred after 8-time and before 7-time.)

Print Cycles. These hubs are similar in use to card cycles of the Type 407. A pulse is emitted from these hubs during every machine cycle of the printer.

Sense Exits. Exits 1 through 10 are addressed by the numbers 241 through 250, respectively. By programming a PSE instruction with one of these addresses, an impulse is made available at the corresponding exit hub. This pulse can then be used to pick up pilot selectors. If the exit is wired in this way, the normal use is as follows: If the PSE instruction is given during the hatched portion of the card cycle, the pilot selector is transferred for the duration of the cycle initiated by the WRS instruction that also is given for the same cycle. PSE instructions given at times in the card cycle other than those specified above may have no effect. (Additional uses of sense exits are discussed below under "Carriage Controls.")

Carriage Control

A punched paper tape (the control tape) used in combination with the ten sense exit hubs usually controls the carriage of the printer. (For some simple applications, such as line-by-line printing, the carriage can be directly controlled without the use of the control tape.) In brief, the control tape is used as follows:

The tape is cut to the length of the form to be used (and later glued into a loop to provide for repetitive operation); thus punched holes in the tape correspond to positions on the form. When the carriage is in operation, the tape advances in synchronism with the form. An impulse to a given carriage skip hub (number 4) causes the form to skip until the control tape—and consequently the form—reaches the position where there is a punched hole in the channel (column) corresponding to the impulsed hub (channel 4).

For a detailed description of the above points and for a further description of the carriage manual controls (restore key, space key, and so on), see the *Type 407 Principles of Operation manual.*

The hubs listed below, when impulsed from the carriage control hubs, activate the various carriage skips and spacings. Remember, while reading the descriptions of these hub functions, that an automatic space is *initiated,* but *not* automatically terminated, before each line of printing, except before printing the first line immediately after skipping. Before the first line of printing or in the cycle immediately after a skip, no normal spacing takes place.

Carriage Skips. These hubs are similar to the D hubs of the Type 407 carriage skips. For example, a pulse to carriage-skips hub 1 begins a form skip that stops when the first punch in channel 1 passes under the control-tape read brushes. In general, a hole punched in a given channel of the tape stops the form at a predetermined position after a pulse to the corresponding carriage-skips hub has started a form skip. The ten channels in the tape (with the ten corresponding carriage skips) provide for an almost unlimited number of combinations of such skips. Because tape length and form length are equal, it is easy

to make the punches in the tape correspond to the predetermined positions on the form.

By wiring sense exits to carriage skips and by punching the various channels in the tape to correspond to various sequences of printing on the forms, it becomes possible for stored programs to maintain a flexible control over the printed output.

Short Skip. These hubs resemble the short-skip hubs of the Type 407. The short-skip hubs provide for skipping with no interruption of printing. The hubs can be used whenever an overflow of less than one inch or a regular skip of less than two inches occurs. Any impulse used to initiate a short skip (e.g., a sense-exit impulse to cause a skip of less than two inches) should be wired first to a short-skip hub and from there to its ultimate destination (a carriage-skip hub). As a result of such wiring, printing continues at the normal rate of 150 lines per minute during short skips.

Space-Sel (*selective space*). These hubs are similar to the selective space hubs of the Type 407. When connected, the two selective space hubs allow spacing of less than seven lines to be selectively controlled by punches in channel 11 of the control tape. The action is such that, before each line of printing, spacing is automatically started; a punch in channel 11 then stops the spacing. For spacing of less than three lines it is necessary only to connect the selective space hubs and punch the desired positions in the control tape. For spacing of more than three lines (but less than seven lines), it is also necessary to impulse the space suppress and extra space hubs from print cycles. (Space suppress and extra space are discussed below.)

Extra (*extra space*). These hubs are similar to the extra-space hubs on the Type 407. Usually these hubs are used with the space 1 or the space 2 hubs. When space 1 is wired and an extra-space hub is impulsed (by a print cycles pulse, for instance), an extra single space results. With space 2 wired, an extra double space results.

SUP. This hub resembles the space suppress hubs of the Type 407. If one of these hubs is impulsed (by a print cycles, for instance), all normal spacing is suppressed during the cycle in which the hub was impulsed.

NP. This hub is similar to the non-print hubs of the Type 407. The NP hubs suppress both printing and spacing, regardless of other control panel wiring, for the cycles in which they are impulsed by a print cycles pulse.

The three types of hubs to be described below may be used to control the carriage. Note that impulses available from these hubs may be directed through various selectors to provide controlled variations in form spacing from print cycle to print cycle.

Sense Exits. A given one of these hubs emits a pulse when a PSE instruction, with the appropriate address, is executed. (See the previous discussion of sense exits in this section.) These hubs can be wired to carriage-skips hubs, thus enabling the stored program in the calculator to control form spacing. When the sense exits are to be used for this purpose, give the corresponding PSE instruction *immediately* after the WRS instruction that starts the print cycle.

OVFL. This hub is similar to the Type 407 overflow hub. It emits a pulse whenever a punch in channel 12 of the control tape passes the control tape reading brushes. The pulse emitted by this hub lasts through the hatched portion of the cycle, after the cycle during which overflow has occurred. (This fact will be of use in discussion of the sense-entry hub below.) The overflow hub is often wired to a carriage-skips hub, thus providing a skip from the position at which channel 12 was punched to the position of the first punch encountered in the channel corresponding to the carriage-skips hub. Usually such wiring is used to overflow to another form after it reaches the bottom of a given form.

SE (*sense entry*). The sense-entry hub is an input hub on the printer control panel. It can be sensed by a PSE instruction with address 240. If, during the execution of this PSE instruction, the sense-entry hub is being impulsed, the PSE instruction skips over the next instruction in the stored program. If the hub is *not* being impulsed when the PSE instruction is executed, the stored program continues without skipping. The sense entry is intended primarily to be used with the overflow hub to inform the stored program of a form overflow. To do this, the overflow hub is connected to the sense entry hub and a PSE instruction is given sometime within the portion of the print cycle in which the overflow hub is emitting a signal. (See above paragraph.)

Spacing

Single, double and selective spacing before printing is carried out by jackplugging the proper space control hubs and, in the case of selective spacing, punching the paper tape for the spacing desired. When irregular spacing is required, the selective space control hubs are used and the 11 channel of the tape is punched. An extra single, double or selective space may be obtained after printing by impulsing the extra space hub after 5°.

Spacing before printing can be suppressed by impulsing the space suppression hub. For single, double or triple line spacing, it is usual to start carriage spacing before printing. However, if the distance to be spaced exceeds three line spaces, there is not sufficient time to complete the spacing before printing. To have the maximum time between start of spacing and printing time, spacing is changed to an after-printing operation by wiring print cycles to both extra space and space suppress.

Wiring the space suppress only would suppress all before-print spacing. Wiring the extra space causes spacing after printing. A maximum of six lines can safely be spaced after printing without interlocking the 716. *Note:* There is no provision made to interlock the machine and carriage during a spacing operation.

CATHODE RAY TUBE OUTPUT RECORDER

THE TYPE 740 CRT Output Recorder permits the display of output information in the form of spots on the face of two cathode ray tubes. (CRT Display Unit and CRT Recording Unit). Digital information in core storage is automatically converted to analog information which is displayed in the form of intensified spots on the CRT's. The stored program controls the CRT unit.

A camera may be attached to the smaller of the two tubes, so that a permanent film recording may be made of whatever appears on the face of that CRT. The larger CRT permits immediate visual observation. Both CRT's contain the same display. The conversion from digital to analog data, the display of spots on the face of the tubes, and the operation of the camera are all done at electronic speeds under control of the calculator.

Uses

Curve Displays. A curve (or curves) may be generated as a display of successive spots on the faces of the CRT's. Each spot is represented in core storage by a pair of rectangular coordinates. Definition: An $m \times n$ raster means that it is possible to display as many as n horizontal and m vertical lines on the face of the CRT. Thus it is possible to display mn spots on the face of the CRT. An example is the sine curve shown in Figure 39.

Assuming the binary point of the coordinates to be immediately to the right of the last binary digit, the ranges of the coordinates of a point are, respectively,

$$0 \leq X \leq 1023$$
$$0 \leq Y \leq 1023$$

Thus the upper bound of points plotted on the CRT is 1023 and the origin is in the lower left-hand corner. There is a maximum of 1024 positions in the X and Y direction, giving a raster size of 1024×1024 and a minimum increment of 1 between successive points in the X or Y direction.

After a curve has been displayed, and perhaps recorded by the camera, the programmer may immediately display another curve. This second curve may also be recorded by the camera, and so on.

Alphamerical Characters. These characters may be displayed very rapidly on the faces of the CRT's. See "Plotting Alphamerical Characters."

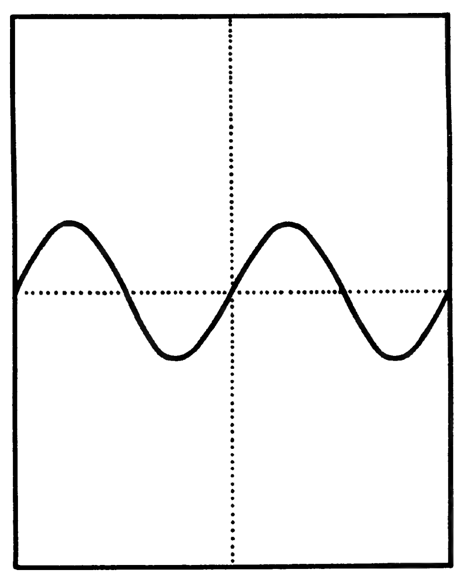


FIGURE 39

Real-Time Problem. "Real time" means the processing of data in synchronism with a physical operation which is actually taking place while the data are being processed. The results of the data processing are used to control the physical operation itself. Examples of this are tracking aircraft and air traffic control.

Simulation Problems. In this type of problem, a physical model and the conditions to which this model may be subjected are all represented by mathematical formulation. An example of this is simulating aircraft performance under structural variations, which makes it possible to test the performance of newly designed aircraft without constructing them.

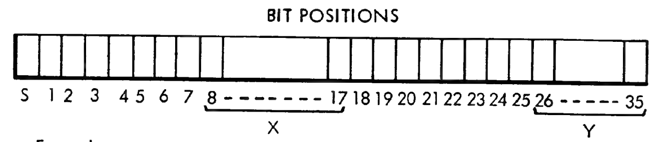
Coordinate Display

Ten binary digits representing the horizontal coordinate (abscissa) of a point are stored in a word of core storage along with ten binary digits representing the vertical coordinate (ordinate) of that point. The display instructions cause the conversion of these digital coordinates to proportional voltages which position the electron beam at the proper point on the cathode ray tubes. As many points as desired may be plotted, one at a time, keeping in mind that the upper bound of the display is 1023.

The instruction WRS 0024 selects the CRT unit in preparation for a display. The display equipment remains selected until some other input-output component (tape, drum, printer, punch, card reader) is selected. The WRS instruction execution takes 24 μs. It need be executed only once in a program unless the CRT unit has been disconnected because of the selection of another input-output component.

After the CRT unit has been selected by the WRS instruction, execution of the CPY instruction causes a spot to be displayed on the face of the two CRT's. (The CPY instruction may be executed *immediately* after the WRS instruction.) The address part of the CPY instruction is the address of the location of a word containing the X and Y coordinates of a point to be displayed. The X coordinate occupies bit positions 8-17, and the Y coordinate occupies bit positions 26-35. See Figure 40.

The instructions shown in Figure 40 would display the point (512, 512) on both CRT's at the center of the area covered by the raster. The coordinates of the



Example:

0200	WRS 0024	1300 | +001000001000 (octal integer)
0201	CPY 1300	

FIGURE 40

point may be seen more clearly by inspecting the binary equivalent of the octal integer in storage location 1300. Bits 8-17 (X coordinate) and 26-35 (Y coordinate) are treated as unsigned numbers. Thus, only the first quadrant of the finite Cartesian plane may be plotted unless the axes are translated. See "Translation of Axes."

There is a minimum delay of 140 μs between the execution of successive CPY instructions. During this time the calculator is free to execute any desired instructions except input-output instructions which disconnect the CRT unit. There is no maximum delay between the execution of successive CPY instructions. The CPY execution takes 48 μs.

NOTE: In the example above, the copy loop would exclude the WRS instruction, since this need be given only once in a program unless the CRT unit becomes disconnected.

Execution of the CPY instruction causes deflection information to be transferred from core storage to the CRT unit buffer registers through the MQ.

Film Recording

The instruction PSE 0024 advances the film in the camera attached to the smaller CRT. Advancing the film exposes a new frame. The frame previous to this new one contains everything that appeared on the face of the CRT since the last PSE 0024 was executed (provided the camera is being operated in a normal fashion). Advancing the film does not affect the shutter; the shutter always remains open. If a display is attempted while the film is being advanced, the display will be delayed until the film is completely advanced.

There is a delay of about 500 ms between a PSE 0024 and a CPY instruction; however, during this time any other instructions may be executed. The PSE instruction execution takes 24 μs.

Spot Intensity

A spot may be programmed on the CRT used for photographic purposes with either one of two intensities, depending upon the contents of position S of the word containing the coordinates of that point. A 1 in the sign position yields a greater intensity than a 0. This gives the programmer an easy method for distinguishing some points (every fifth point, for example) from others.

Axes Generation

A 1 in position 1 of a word containing the coordinates of a point causes a horizontal line to be traced from the selected point to the right edge of the raster. A 0 inhibits the generation of this line.

A 1 in position 2 of a word containing the coordinates of a point causes a vertical line to be traced from the selected point to the upper edge of the raster. A 0 will inhibit the generation of this line. A 1 in both positions 1 and 2 causes a diagonal line to be generated at about a 45° angle.

Data Manipulation

The instruction STP Y may be used to store the contents of positions P, 1, 2 of the accumulator in positions S, 1, 2 (intensity and axes bit positions) of a word that contains the coordinates of a point to be displayed.

The instruction STD Y may be used to replace the X-coordinate in register Y by the X-coordinate in the accumulator, and the instruction STA Y may be used to replace the Y-coordinate in register Y by the Y-coordinate in the accumulator.

Translation of Axes

The nominal axes may be translated so that their intersection (origin) may be moved from the lower left-hand corner to any other position on the face of the CRT by means of simple translation formulas.

EXAMPLE: To translate the axes so that the origin is at the center of the area covered by the raster, use the translation formulas.

$$X_n = 2X_0 - 1024$$

$$Y_n = 2Y_0 - 1024$$

where (X_n, Y_n) are coordinates of a point with respect to the new position of the axes and (X_0, Y_0) are coordinates of that point with respect to the original position of the axes.

Floating-Point Numbers

If the CRT's are to be used for plotting points whose coordinates are floating-point numbers, the characteristics of these numbers should be equalized before an attempt is made to plot their fractional part. It may be that in exceptional cases this will not be necessary. However, ordinarily it will be necessary to equalize so that the coordinates of the plotted points will have the same "weight." Example:

ORIGINALLY	CHARACTERISTICS EQUALIZED
$X_1 = .5 \times 2^6$	$X_1 = .5 \times 2^6$
$Y_1 = .4 \times 2^6$	$Y_1 = .4 \times 2^6$
$X_2 = .2 \times 2^7$	$X_2 = .4 \times 2^6$
$Y_2 = .3 \times 2^7$	$Y_2 = .6 \times 2^6$

Plotting Successive Distinct Points in the *X* or *Y* Direction

In order to plot successive distinct points in the *X* or *Y* direction, one must take into account the spot size diameter. The spot size diameter is such that if two spots are about externally tangent, their centers must be 4 units apart. Thus, even though the raster size is 1024 × 1024, one can get only 256 distinct spots on a line which is horizontal or vertical. A very smooth line may be plotted by selecting an increment of 1 instead of 4.

Plotting Alphamerical Characters

An alphabetic or numerical character can be plotted point by point using a suitable program. The size of the character is determined by the type of pattern selected. See Figure 41.

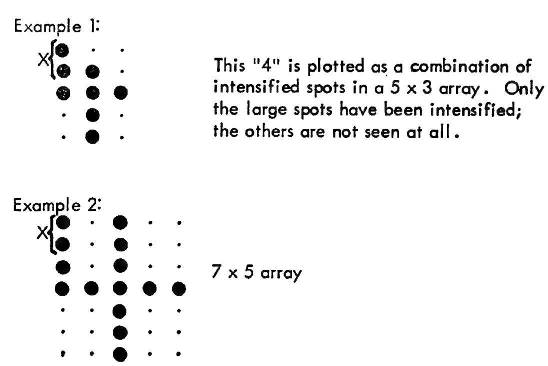


FIGURE 41

The distance X between successive spots is left to the discretion of the programmer. (Refer to *distinct* spot separation discussed previously.)

If the 35 spot positions of the character "4" of Example 2 are made to correspond to 35 binary places of a word in core storage, then a character may be represented by that word at a predetermined location in storage (table look-up). The sign bit of that word may be considered to be positive. The remaining bits signify a display or no display of a spot, depending on whether the bit is a 1 or a 0, respectively. The first column of the character is represented by the first group of 7 bits (following the sign bit), the second column by the second group of 7 bits, the third column by the third group of 7 bits, and so on. The first bit of each group may correspond to the lowest position of each column, and succeeding bits will move up the column, one bit at a time. The character "4" is represented in the table by the binary word and appears on the CRT as the "4" outlined in Figure 42.

Principal Components and Their Operation

DISPLAY UNIT

The display unit provides an immediate display for visual purposes on a 21-inch CRT. The persistency on the face of this tube ranges from two seconds in a brightly lit room to 20 seconds in a dimly lit room, so that a generated curve may be seen in its entirety. The accuracy of this CRT is 1%; that is, if a point with the same coordinates were to be displayed twice, the successive spots would lack coincidence by no more than 1% of full scale.

RECORDING UNIT

The recording unit provides an immediate display on a 7-inch CRT for photographic purposes. The persistence is only a few microseconds, or long enough to be recorded on film by the camera attached to the 7-inch CRT. The position stability of this tube is better than 0.1% and it has a maximum error of 0.5% positional accuracy.

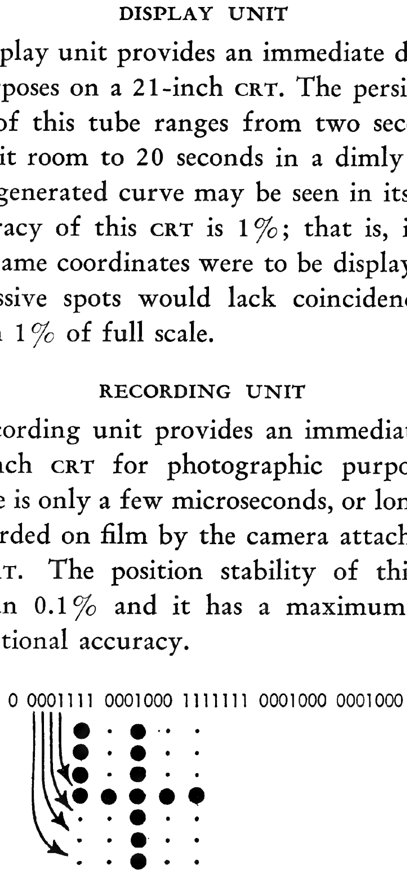


FIGURE 42

The buffer registers in the recording unit make it possible to minimize the amount of calculator time required for displays. The calculator time consumed is only that time necessary for information to be transferred from the calculator to the recording unit buffer registers. The calculator is then free to continue while the information within these buffer registers is automatically converted to analog data and displayed as a spot.

CAMERA AND ITS ASSOCIATED EQUIPMENT

A 35 mm. pulse-operated camera is provided with the recording unit to photograph the display on the 7-inch CRT. Associated equipment permits the camera to function in conjunction with the recording unit under the control of the calculator. Film of exposure index ASA 200 should be used. The film is wound, emulsion-side in, on a wooden core 1¼ inches in diameter with a 7/16 inch center hole. The camera film magazine has a capacity of 100 feet of perforated or unperforated film.

Users should practice loading the magazine in daylight with 3 to 5 feet of film which may be cut from the regular film. One method of loading the film is described below:

1. Place the magazine on a soft cloth on the loading room table with the motor down, cover facing up and the aperture plate forward. Remove the film-securing clip from the take-up hub, 1 in Figure 43, and place it on the cover. Withdraw the dark slide about ¾ of the way. You are now ready to load film.

2. Pick up the roll of film in the right hand in such a manner that the film being unwound is on the 9 o'clock side. Unroll about 12″ of film and place the rest of the roll on the film slide post 2. Keep the 12″ leader in the left hand and with the right hand start at the roll and pass the film back of the guide roller 3, between the pressure plate and the magazine plate, then around the metering roller 4 and back to the take-up hub at 1. Be sure that the film comes around the take-up hub on the right-hand, or 3 o'clock side. Now place the fingers in the film aperture and press in and down. This seats the film on the bottom of the magazine. Check to see that the film edges are even over the tops of both rollers 3 and 4, and down as far as it will go on the bottom of the magazine. Take the film-securing clip from the top of the cover and snap it around the film take-up hub at 1. The cover may now be placed on the magazine

(be sure that the front edge fits properly in the slot on the aperture plate) and the screw tightened. Also, return the dark slide to a normal position.

3. Do not pass the film around the cover support stud. The entire operation will approach 30 seconds, but the operation should not be hurried.

4. When the magazine is first placed on the camera, the motor will run for a short time to set the metering roll. Be sure to remove the dark slide before taking pictures. The magazine motor and related parts should be serviced and oiled with the finest watch oil once a year or at 100,000 pictures, whichever occurs first.

5. To remove a short length of film, cut the film between rollers 3 and 4, with scissors, and rethread the camera with another take-up spool.

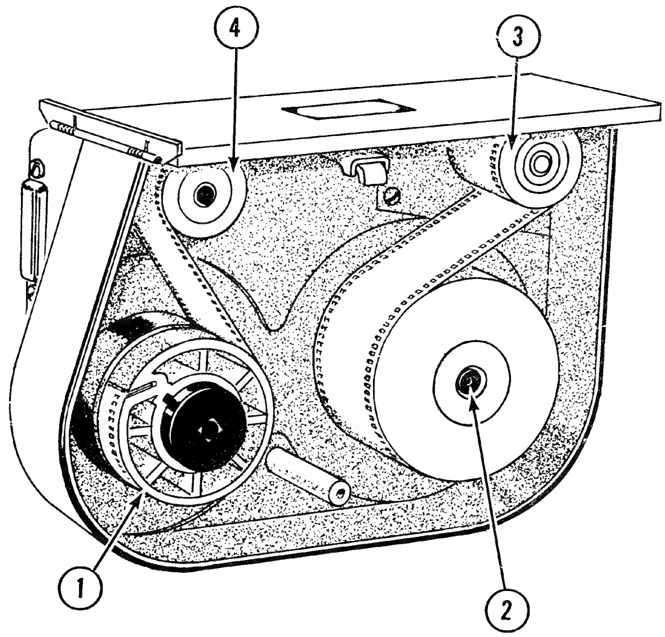


FIGURE 43

THIS section of the manual describes available peripheral equipment using standard 704 input-output units. Tape units can be connected to card punches, card readers, and printers to provide independent machines that can perform many operations not requiring the logical ability of the 704.

The Type 719 or Type 730 Printer with a Type 760 Printer Control Unit and a Type 727 Tape Unit may be used for peripheral printing. This operation is described in a preliminary edition of the manual of operation for the 719 and 730 printers.

CARD-TO-TAPE CONVERTER

THE TYPE 714 card reader and its Type 759 control unit may be connected by cables to a tape unit to record on magnetic tape data from IBM cards.

By control panel wiring it is possible to transcribe the entire card on tape or to select any combination or arrangement of fields. The sensing of the record storage mark causes an inter-record gap on the tape. Unpunched card columns preceding the record storage mark are written as blank characters on tape. Characters may also be emitted from the control panel or the grouping feature may be used to convert two card records to a single tape record.

Operation

To accomplish a card-to-tape conversion, a tape unit is connected by cable to a card reader and the card reader control unit. The tape is properly loaded in the tape unit and the door is closed. Depressing the tape unit load-rewind key feeds the tape into the vacuum columns to be taken to the load point. Depressing the start key then turns on the ready light.

When the ready status has been established in the tape unit, the card reader may be operated. Place cards in the hopper and press the start key once to run in the cards. A second depression of the start key causes the card-to-tape conversion to begin.

When the end of the tape is reached, the operation stops. Remove the cards from the hopper and press the start key to record the last two cards in the feed on tape. A tape mark is automatically recorded after the last card. Rewind the tape by pressing the load rewind key. Remove the tape from the columns by pressing the unload key and taking off the reel. Repeat the normal starting procedure to resume the operation.

When the end of the card file is reached, end the operation by pressing the start key and recording the last two cards in the feed. A tape mark is automatically recorded after the last record. If desired, an additional file of information can be written by loading additional cards without rewinding the tape.

Recording

All standard punched card characters are recorded on tape as indicated by the character code chart (Figure 44). The following card punches are recorded as indicated:

CARD	TAPE
Zone 11, numerical 0	$\bar{0}$
Zone 12, numerical 0	$\overset{+}{0}$
Zone 0, numerical 2 and 8	Record Mark
Zone 12, numerical 5 and 8	Group Mark

Checking

During a card-to-tape operation, checking consists of the following:

1. The cards are read at two brush stations. At the first brush reading, the number of holes in each horizontal row of the card is determined to be odd or even. This information is stored in twelve binary triggers. The reading of the card at the second set of brushes reads the card into record storage. When data are read out of record storage, the number of holes in each horizontal row of the card is again checked for odd or even number. This number is stored in another set of twelve binary triggers. These two sets of triggers are compared to insure correct card reading.
2. After the card record has been recorded on tape, the tape is backspaced and read for a lateral check for each character and longitudinal row check for each record.

Whenever the control panel is wired to control information going into record storage, it is also necessary to wire it into the check entry.

	CHAR.	CHAR.	CHAR.		
	& 0 11 0000	- 1 10 0000	Blank 1 01 0000		
ALPHABETIC	A 1 11 0001	J 0 10 0001	/ 0 01 0001	1 1 00 0001	NUMERICAL
	B 1 11 0010	K 0 10 0010	S 0 01 0010	2 1 00 0010	
	C 0 11 0011	L 1 10 0011	T 1 01 0011	3 0 00 0011	
	D 1 11 0100	M 0 10 0100	U 0 01 0100	4 1 00 0100	
	E 0 11 0101	N 1 10 0101	V 1 01 0101	5 0 00 0101	
	F 0 11 0110	O 1 10 0110	W 1 01 0110	6 0 00 0110	
	G 1 11 0111	P 0 10 0111	X 0 01 0111	7 1 00 0111	
	H 1 11 1000	Q 0 10 1000	Y 0 01 1000	8 1 00 1000	
	I 0 11 1001	R 1 10 1001	Z 1 01 1001	9 0 00 1001	
	Plus Zero 0 0 11 1010	Minus Zero 0 1 10 1010	Record Mark 1 01 1010	Numerical Zero 0 0 00 1010	
SPECIAL CHAR	. 1 11 1011	$ 0 10 1011	, 0 01 1011	# 1 00 1011	
	¤ 0 11 1100	* 1 10 1100	% 1 01 1100	@ 0 00 1100	
	Group Mark 1 11 1101			Tape Mark 0 00 1111	

FIGURE 44

Grouping

This feature permits the grouping of two card records into one tape record with the following restrictions:

1. The maximum tape record is 92 characters, the size of the record storage unit.
2. No more than 46 characters from a single card.
3. The same card columns must be used from the two card records.

Controls

The tape unit controls are described in the section on magnetic tape units. The Type 714 card reader controls include those described in the section on the Type 711 card reader and, in addition, further controls for the card-to-tape operation are described below. The start, stop, and feed keys and the ready, select, and feed check (card feed stop) lights operate identically to the corresponding keys and lights on the Type 711 card reader.

Backspace Key. The backspace key, when depressed, backspaces the tape one record for each depression. It permits rewriting of records that have been recorded incorrectly.

Reset Key. Depressing the reset key resets the check circuits and allows normal operation.

Read-Check Light. The read-check light goes on when a card reading error is detected.

Write-Check Light. The write-check light is turned on when a writing circuit error is detected.

TAPE-TO-CARD CONVERTER

THE TYPE 722 card punch and the Type 758 control unit may be attached directly by cables to a tape unit to transcribe magnetic tape records written in the BCD mode to IBM punched cards. As many files as desired may be recorded from each tape.

Records of 78 characters or less may be punched at a speed of 100 cards per minute. Records longer than 78 characters may cause an error indication and should not be used. Information is punched in the cards in the same order that it is read from tape.

Operation

Prepare the tape unit for operation by properly loading a reel of tape on the tape unit and closing the door. Pressing the load-rewind key brings the tape automatically to the starting point. Subsequently pressing the tape unit start key turns on the ready light.

When the ready status has been established in the tape unit, place the cards in the card punch hopper and press the start key twice to begin the operation.

When a tape mark is sensed, the operation is stopped. Punch the next file by pressing the start key, or rewind the reel at this time by pressing the load-rewind key. Press the unload key to remove the tape from the vacuum columns. The reel may then be removed from the unit. Load another reel onto the unit and feed it to the starting point by pressing the load-rewind key. Press the tape unit start key and then press the card punch start key twice to resume operation.

After the last reel of tape has been recorded, pressing the feed key on the card punch removes the cards from the card punch hopper and runs the remaining cards out of the feed.

Checking of Reading

Information read from tape is given a character-by-character code check. In addition, a longitudinal check for an even number of ones in each of the seven tape channels is made for each record to determine whether a single one has been changed in any channel in reading a record. A failure detected by either of these two methods stops the machine before it punches the erroneous record. It also turns on the read-check light on the control unit.

To reread the record in question after detecting a reading error, press the backspace key on the card punch to backspace the tape one record. Pressing the restart key on the card punch turns off the error indication and causes the card punch to execute a clearing cycle. On the clearing cycle a card is fed but not punched and record storage is cleared. The operation is then automatically resumed with the same tape record being reread into record storage.

If, upon detecting a reading error, the record is not to be reread, press the restart key to turn off the error indication and resume operation. In this event no clearing cycle occurs and the record in record storage is punched.

Checking of Punching

The punched cards are checked at a brush station one card cycle after being punched. The check is an odd-even horizontal row count of the holes in the card matched against a similar row count of the tape record sent to record storage.

A discrepancy in a card is indicated by the punch-check light which turns on after the following card has already been punched and the second record following has already been read into record storage. Therefore, to reread the record in error, press the backspace key on the card punch three times to backspace the tape the necessary three records. Pressing the restart key on the card punch then causes the error indication to be turned off, a card to be fed but not punched, the record storage to clear, and the operation to resume. Therefore, a blank card is fed and two cards are repunched. Remove the blank card and the two cards preceding it.

Punching

The numerical, alphabetic and special characters are punched in the standard IBM card code. The following characters are punched as indicated.

TAPE CHARACTER	CARD PUNCHING
$\bar{0}$	Zone 11, numerical 0
$\overset{+}{0}$	Zone 12, numerical 0
Record Mark	Zone 0, numerical 2 and 8
Character Code Error	Not punched
Group Mark	Zone 12, numerical 5 and 8

Controls

The tape unit controls are described in the section on magnetic tape units. The Type 722 card punch controls include those described in the section on the Type 721 card punch and, in addition, further controls for the tape-to-card operation are described below. The start, stop, and feed keys and the ready, select, and feed check lights operate identically to the corresponding keys and lights on the Type 721 card punch.

Backspace Key. This key, when pressed, backspaces the tape one record for each depression. Records that have been read erroneously may be backspaced and reread.

Punch-Check Light. This light is turned on when a discrepancy is revealed by the check performed on punching.

Read-Check Light. This light is turned on when a character code error is detected in reading a tape record.

Restart Key. Pressing the restart key resets the error indication and resumes the punching operation after an error has been detected and the error indication has been turned off. If the tape has been backspaced to reread the records in question, pressing the restart key causes the card punch to execute a clearing cycle, after which the operation is automatically resumed.

TAPE-CONTROLLED PRINTER

THE TYPE 717 printer and the Type 757 control unit may be connected by cable to a tape unit to provide a method of direct printing of information from magnetic tape. Information is printed in the same order in which it is read from tape at a speed of 150 lines per minute. As many files as desired may be printed from each tape.

Records of 120 characters or less may be printed. Records of 120 characters may be read with carriage control switch at PROGRAM where the first character is not printed. Tape records in excess of 120 characters may cause a machine stop or an error indication and therefore should not be used.

Carriage Control Switch

The carriage control switch on the tape-controlled carriage may be set to single space, double space, or program.

When the switch is set to single space, form spacing is six lines to the inch. When the switch is set to double space, spacing is three lines to the inch. Under either setting, print-wheel one prints the first character in the record, while each successive character is printed by wheel two, three, and so on. The carriage tape controls the ejection and spacing of the form. Channel 1 is the restore or home position of the form and channel 12 is the overflow or eject position. When channel 12 is sensed while printing a line, the form is automatically ejected to channel 1. Controlled skipping to other positions of the form is not possible unless the carriage switch is set to PROGRAM.

With the switch set to PROGRAM, the skipping of the form is under the control of the first character in the record and ejection is not automatic. This character is the carriage control character and is not printed. Print-wheel one prints the second character in the record, and successive characters are printed by print-wheel two, three, and so on. The following characters may be used to control skipping:

Suppress space	& (ampersand)
Single space	b (blank)
Double space	0
Skip to channels 1-9	1-9
Short skip to channels 1-9	J-R

When preparing a tape for peripheral tape-to-printer operation with the carriage control switch set to PROGRAM, the printing lines must be counted to simulate carriage overflow. The proper control character is inserted as the first character of the tape record.

Operation

The tape unit is prepared for operation first by putting in the proper reel and closing the cover. Depressing the tape load-rewind key brings the tape automatically to the starting point. Depressing the tape unit start key then turns on the ready light on the tape unit.

The printer is made ready by inserting the proper paper form, by inserting a carriage tape if any is required, by restoring the carriage to channel 1, and by setting the carriage control switch. Pressing the printer start key once causes a clearing cycle. A second depression starts the printing operation.

When a tape mark is sensed, the operation is stopped. Print the next file by pressing the start key or rewind the reel at this time by pressing the load-rewind key. Press the unload key to remove the tape from the vacuum columns. The reel may then be removed from the unit.

Checking of Reading

Information read from tape is given a character-by-character, even-count, character code check. In addition, a longitudinal check for an even number of bits is made for each of the seven tape channels for each record. This determines whether a single bit has been changed in any channel in reading a record. A failure detected by either of these two methods stops the machine before it prints the erroneous record. It also turns on the read-check light on the printer.

To reread a record found to be in error, press the backspace key on the printer to backspace the tape one record. Pressing the restart key on the printer resets the error condition and causes the printer to go through a clearing cycle. On the clearing cycle, the erroneous record is removed from record storage and no printing or carriage spacing occurs. The operation is then automatically resumed with the same tape record being reread into record storage.

If it is not necessary to reread the record when a reading error is detected, press the restart key to resume operation. In this event, no clearing cycle occurs and the record in record storage is printed.

Checking of Printing

The printing of information is checked by comparing the print-wheel echo impulse count against the information sent to the printer record storage. This is a row count check of the numerical portions of all characters. A discrepancy revealed by this check stops the machine after it prints the record in error. It also turns on the printer check light.

Should a printing error be detected, the erroneous record will have been printed and the next record will be in record storage at the time the machine stops. Therefore, to reprint the erroneous record, the tape must be backspaced two records by two depressions of the backspace key. Pressing the restart key resets the error indication and causes a clearing cycle

after which the two tape records will be reread and normal operation will be resumed. If the record in error is not to be reprinted, pressing the restart key resumes operation.

Printing

All the characters in the character code chart (Figure 44) are printed as indicated except for the following:

CHARACTER	PRINTED
$\overset{+}{0}$	&
$\overset{-}{0}$	—
Record mark	Blank
Character code errors	Blank
Group Mark	⌑

Controls

The tape unit controls are described in the section on magnetic tape units. The Type 717 printer controls include those described in the section on the Type 716 printer and, in addition, further controls for the tape-to-printer operation as described below. The start and stop keys, ready, select, and form stop lights, and the platen clutch, restore, stop, and space keys on the carriage operate identically to the corresponding controls on the Type 716 printer.

Carriage Control Switch. Automatic single or double spacing will occur when this switch is set to single or double space. If the switch is set to program, spacing is controlled by the first character in each record. This character is not printed.

Form Control Key. The printer does not stop when the form stop contact closes. After the form stop has closed and channel 1 on the carriage tape is sensed, the printer stops before printing on the next form and the form stop light is turned on. This indicates that the carriage has moved from one form to another. The operator now has two options governed by the form control key.

1. Manually feed new forms of paper into the carriage, thereby opening the form stop contact. Pressing the form control key then turns off the form stop light and allows the operation to continue.
2. Press the form control key to turn off the form stop light and allow the printer to continue until the closed form stop contact and channel 1 condition again stops the printer.

Thus one or more forms may be printed before finally stopping the printer and inserting additional paper.

Backspace Key. Pressing the backspace key causes the tape involved in the tape-controlled printer operation to be backspaced one unit record. Successive depressions of this key will cause the tape to be backspaced one record for each depression.

Restart Key. Pressing the restart key resets the error indication and resumes the printing operation after an error has been detected.

If the tape has been backspaced to reread the record in question, depression of the restart key causes a printer clearing cycle, after which operation will be continued.

Write-Check Light. If a discrepancy is detected by the printer echo check, the machine stops after printing the record and the write check light turns on.

Read-Check Light. If a character code error is detected in reading a tape record, the machine stops after reading that entire record and the read check light turns on.

THE 704 can execute a program only if it is an *absolute* program; that is, if all words, whether instructions or data, have been assigned definite locations in storage, and all quantities in the program depending upon the allocation of storage have been assigned definite numerical values accordingly. But it is usually difficult to write a program in this form, because an intelligent assignment of storage space cannot be made until *after* the program is written and the demands for space are known. Therefore, some non-absolute system of program writing is usually used.

The programming system recommended for the Type 704 is called *symbolic programming*. The program is written in terms of symbols denoting the storage locations (as yet unknown) of all the words to which the program must make reference. When this *symbolic* program is completed, the programmer can decide how he wishes the program to be fitted into storage. The symbols now take on their definite numerical values, and the symbolic program becomes absolute. These final steps, of assigning values to the symbols in accordance with the programmer's wishes, and replacing the symbols throughout the program by their values to produce the absolute program, are carried out by the symbolic assembly program NY AP 1. This assembly program accepts the symbolic program in the form of standard IBM cards, key-punched from the programming sheets, together with additional cards specifying the programmer's wishes about storage allocation; it produces the absolute program in the form of binary punched cards suitable for subsequent loading into the computer, together with a printed listing of the program in both symbolic and absolute form. For details, consult the NY AP 1 write-up.

The following points are of special interest to the programmer.

1. It is not necessary to assign a symbol to a location until you find that you must refer to that location. In practice, you never refer to the majority of the words in a program. Consequently, in the finished symbolic program only a small fraction of the locations are given symbols.

2. When you must invent a symbol to assign to a location, it may be any six of the 47 characters expressible in the IBM code (e.g., H, 3, or $). Thus, symbols of high mnemonic value may be used; for example, TRY A, B and X SUB 2 might be assigned to the locations of any instruction and a word of data.

3. Use NY AP 1 to refer to the second, third, or the *n*th location before or after a location which has already received a symbol. Notice that on the programming form both the address and decrement fields are divided into a symbolic and an absolute part. Suppose that the symbolic part of the address of an instruction is ALPHA and the absolute part is 4. Then in the final absolute program, if ALPHA has become, say, 1000, the address of this instruction will be 1004. This provision is particularly convenient in connection with a block of data, since a single symbol will serve for references to any of the words in the block.

4. Moreover, if the symbolic part of the address had been left blank (instead of containing ALPHA), the address in the absolute program would have become 4 (instead of 1004). Therefore, if the absolute value of an address or decrement is known, it should be written in the absolute part and the symbolic part left blank.

The following programming examples appear in symbolic programming form, exactly as they would be written for assembly by NY AP 1.

N-Way Branch of Control

In Figure 45, suppose that some quantity *P* may have any of the values 1, 2, or 3 and that control is to be transferred to the location P IS 1, P IS 2, or P IS 3.

Notice that in the program P does not mean the quantity *P*, but the symbol for the storage location where P is to be found. Until it is well understood, this identification of symbol with quantity can cause difficulty, but, once understood, its use is a major advantage of symbolic programming. This principle may be extended to an *n*-way branching.

Identification	Location	Operation Code	Address Symbolic	Address Absolute	Tag	Decrement Symbolic	Decrement Absolute	Comments
		LXA	P		A			Load index register A with P
	BRN	TRA	BRN	4	A			TRA to appropriate TRA below
		TRA	P IS 3					Transfer
		TRA	P IS 2					to the
		TRA	P IS 1					correct location

FIGURE 45

Normalizing an Unnormalized Floating-Point Number

Normalize unnormalized floating-point numbers as shown in Figure 46.

Floating a Fixed-Point Number

A fixed-point number, whose binary point is Q places to the left of its right-hand end, is to be converted into normalized floating-point form. No assumptions are to be made about the number of leading zeros in the fixed-point word, and no binary digits are to be discarded unnecessarily. See Figure 47.

The program proceeds as follows:

1. If the fixed-point number does not have room for the characteristic (does not possess at least eight leading zeros), it is shifted until there is room. The number of places shifted, S, is counted in an index register.
2. The characteristic, which equals $128 + S - (Q - 27) = 155 + S - Q$, is computed and a floating-point number is formed.
3. If no shifting was done earlier ($S = 0$), the number may be unnormalized. In this case, use the normalizing procedure of the preceding example.

The only requirement upon S and Q is that $0 \leqq 155 + S - Q < 256$. A test could be built into the program to cause a HALT if this condition is violated.

Fixing a Floating-Point Number

A floating-point number is to be converted into fixed-point form with the binary point Q places to the left of the right-hand end (Figure 48).

The "fixer" is a floating-point number, with fractional part zero and characteristic $C = 163 - Q$. This program assumes that Q is known at the time of writing, and $163 - Q$ may, therefore, be entered in the fixer at that time. (If Q is not known in advance, a minor routine to compute the fixer could precede the present program.) The FXD (fixed decimal) in the operation field of fixer instructs NY AP 1 to convert 139 from decimal to binary and place it in the word with its binary point 27 places to the left of the right-hand end, that is, in the characteristic field for a floating-point number.

Suppose that the floating-point number to be fixed has a characteristic C. Examination of the program shows that everything will be accomplished correctly, provided Q and C satisfy $163 - Q \geqq C$. If, however, Q and C do not satisfy this relation, probably the programmer has made an error; for this combination demands that leading non-zero bits be lost. The present program does not cause them to be lost, but instead produces the fixed-point number with a Q less than the desired Q. A test for $163 - Q < C$ could easily be incorporated to give warning that this condition has occurred.

Identification	Location	Operation Code	Address Symbolic	Address Absolute	Tag	Decrement Symbolic	Decrement Absolute	Comments
		CLM						Clear AC
		FAD	UNNOR					Normalize by use of FAD
		STO	NORM					Store result

FIGURE 46

Identification	Location	Operation Code	Address Symbolic	Address Absolute	Tag	Decrement Symbolic	Decrement Absolute	Comments
		LXD	TXH		A			Prepare
		CLM						for program
		LDQ	FIXED					Bring fixed-point word into MQ
		LLS		8				Is there room
	TZE	TZE	PXD					for characteristic
		LRS		1				If not, shift, add 1 to S,
		TXI	TZE		A		1	and ask again
	PXD	PXD			A			Form characteristic
		SUB	Q					and store
		ALS		9				for later
		STO	ERASB					normalizing
		ARS		27				Compose
		LLS		27				floating-point word
	TXH	TXH	STO		A		155	Is normalizing necessary
		FAD	ERASB					Normalize
	STO	STO	FLOTG					Store
	Q						Q	

FIGURE 47

Double-Precision Floating-Point Division

As a final example of programming associated with floating-point numbers, suppose that the quotient A/B is to be formed, where A and B are each stated in double-precision floating-point form; A, for example, is expressed as $A_1 + A_2$, where A_1 and A_2 are floating-point numbers, A_1 is normalized, and A_2 has a characteristic 27 less than that of A_1. The quotient $Q = A/B$ is to be produced in the same form (Figure 49).

The program rests upon the fact that, since B_2 is very much smaller than B_1,

$$\frac{A}{B_1 + B_2} = \frac{A}{B_1} - \frac{A}{B_1} \times \frac{B_2}{B_1}.$$

It evaluates this formula in straightforward fashion:

1. A_1 is divided by B_1 to form the most significant part of A/B_1.
2. A_2 is added to the remainder and the result is divided by B_1 to form the least significant part of A/B_1.

Identification	Location	Operation Code	Address Symbolic	Address Absolute	Tag	Decrement Symbolic	Decrement Absolute	Comments
		CLA	FIXER					Do UFA of floating number
		UFA	FLOTG					and fixer
		RQL		9				Discard characteristic in MQ
		LGL		8				Recover least significant digits
		STO	FIXED					Store
	FIXER	FXD		139		27		This is for case Q=24. 163-24 is 139

FIGURE 48

Identification	Cards	Location	Operation Code	Address Class	Address Symbolic	Address Absolute	Tag	Decrement Class	Decrement Symbolic	Decrement Absolute	Comments
			CLA		A1						Form most significant
			FDH		B1						part of A/B1
			STQ		A/B1 1						and store
			FAD		A2						Form least significant
			FDH		B1						part of A/B1
			STQ		A/B1 2						and store
			CLA		B2						Form
			FDH		B1						Minus
			FMP		A/B1 1						B2/B1 times the most
			CHS								significant part of A/B1,
			FAD		A/B1 2						Add
			FAD		A/B1 1						A/B1,
			STQ		Q1						And
			STQ		Q2						Store

FIGURE 49

3. B_2 is divided by B_1 and the result multiplied by the most significant part of A/B_1 (multiplication by the least significant part of A/B_1 is not necessary because of the order of magnitude of the numbers involved) and the sign is changed.
4. Finally, A/B_1 is added in two stages.

Drum Copy Loop

The 100 consecutive words beginning with location 1000 on drum 301_8 are to be read into the 100 consecutive storage locations beginning with 1STWD (Figure 50).

The O in the address class column of the RDS instruction indicates to NY AP 1 that the absolute part of the address has been stated in octal.

The arrangement LXA, ..., ..., ..., TIX is the simplest and most common form of loop. It is very easy to write; if the instructions in the loop are to be executed n times, load the index register with n and make the address of each instruction tagged with that index register equal to the address it should have on its first execution plus n.

This loop also has the property that it "moves" upward through storage; that is, the effective addresses of the tagged instructions increase as the program is executed.

Example of Loop Writing

An array of 100 quantities C_{ij} is to be computed according to the expression

$$C_{ij} = A_i - B_j \quad \text{for } i > j$$
$$= A_i + B_j \quad \text{for } i \leqq j \quad (i, j = 1, 2, \ldots, 10).$$

The program is shown in Figure 51.

Notice that this type of loop, using TXI to change

Identification	Cards	Location	Operation Code	Address Class	Address Symbolic	Address Absolute	Tag	Decrement Class	Decrement Symbolic	Decrement Absolute	Comments
			RDS	O		301					
			LDA		1000						
			LXA		100		A				
		CPY	CPY		1STWD	100	A				
			TIX		CPY		A			1	
		1000				1000					
		100				100					

FIGURE 50

IDENTIFICATION	CARDS	LOCATION	OPERATION CODE	ADDRESS CLASS	ADDRESS SYMBOLIC	ADDRESS ABSOLUTE	TAG	DECREMENT CLASS	DECREMENT SYMBOLIC	DECREMENT ABSOLUTE	COMMENTS
			LXD		INC J		7				Initialize (make I, J, IJ EQUAL 1)
		STOR J	SXD		COM IJ		B				Store J for comparison with I
			CLA		B1	1	B				CLA B SUB J
		COM IJ	TXL		ADD		A		( )		Is I less than or equal to J
			CHS								If not change sign of B SUB J
		ADD	ADD		A1	1	A				Add A SUB I
			STO		C11	1	C				Store answer in C SUB IJ
		INC J	TXI		INC IJ		B			1	Increase J by 1
		INC IJ	TXI		TEST J		C			1	Increase IJ by 1
		TEST J	TNX		STOR J		B			10	Test J and go back or reset J
			TXI		TEST I		A			1	Increase I by 1
		TEST I	TNX		STOR J		A			10	Test I and go back or go on

FIGURE 51

the index quantities "moves" backward through storage. Hence the A's, for example must be stored in the order $A_{10}, A_9, \ldots, A_1$, and the C's are stored in the order $C_{10,10}, C_{10,9}, C_{10,8}, \ldots, C_{1,3}, C_{1,2}, C_{1,1}$.

Subroutines

Many routines, such as decimal-to-binary conversion, binary-to-decimal conversion, sine, exponential, and so on, can be used repeatedly since they perform basic functions reappearing several times within a single program or in different programs. These are called subroutines because they are subordinate to the main program in which they are used. In general, each large program uses several subroutines in addition to the special logic peculiar to the problem involved. The set of all subroutines used in a computing installation is called a library.

Open Subroutine. An open subroutine is a set of instructions to be integrated into the logical flow of the main program. In using an open subroutine in more than one place in the main program, insert it in each place. In general, the data needed by the open subroutine is placed in the specified locations in the subroutine by the main program. Then the flow of control goes directly into the subroutine from the main program without a transfer of control; control exits from the subroutine directly into the main program again. Hence the open subroutine is sandwiched directly into a program as though it were part of the original coding of the program.

In the following example, the subroutine converts 12 words (72 characters) of BCD information to a card image and prints the line of information on the alphabetic printer. To use this open subroutine, the main program must select the printer and put the first six words to be printed in the storage locations D through $D + 5$. The last six words to be printed must be stored in $D + 32$ through $D + 37$, where D is the symbolic origin for the data. The subroutine transfers control back to the main program which resumes at $D + 55$. Program constants for the subroutine are stored in $D + 23$ through $D + 29$. $D +$ 30, 31 are unused; hence, they may be used for any kind of storage by the main program. This subroutine uses all three index registers (Figure 52).

Closed Subroutine. A closed subroutine may occur several times within one main program, but the set of instructions comprising the subroutine need appear only once in the main program. The transfer of control from the main program to the subroutine takes place from a calling sequence (*basic linkage*). Specify the calling sequence to be used in the subroutine. It varies according to the data needed by each subroutine. For example, if a subroutine for computing the sine of x requires one word of data, the calling sequence is shown in Figure 53.

The subroutine can compute the exit address that takes it back to the main program by using the fact that index C contains the 2's complement of the address of the TSX instruction. Thus the sine program would be as shown in Figure 54.

IDENTIFICATION	CARDS	LOCATION	OPERATION CODE	ADDRESS SYMBOLIC	ADDRESS ABSOLUTE	TAG	DECREMENT SYMBOLIC	DECREMENT ABSOLUTE	COMMENTS
		BCD	LXA	D	26	A			Choose left card image
			LXD	D	26	B			Clear
			CLM						D + 6
			SLW	D	87	3			thru
			TIX	BCD	3	B		1	D + 22
			CAL	D	23				Refresh
			SLW	D	27				column indicator
			LXA	D	24	B			Beginning
			LDQ	D	70	3			of preliminary
			SXD	D	29	B			card image
			CLM						formation
			LGL		2				Place zone part
			PAX			B			in index B
			CLM						Clear accumulator
			LGL		4				Place numerical part
			PAX			C			in index C
			CAL	D	27				Form present
			ORS	D	86	3			column of
			ORS	D	82	5			card image
			ARS		1				Move to
			SLW	D	27				next column
			ANA	D	28				Close loop if word
			TNZ	BCD	10				is not finished
			LXD	D	29	B			Move to next word and
			TIX	BCD	8	B		1	test end of half image
		ABCD	LXD	D	24	B			Or
			CAL	D	72	3			8-4 word
			ORS	D	74	A			and 8-3 word
			ORS	D	80	3			into
			TIX	ABCD	1	B		1	8, 4, and 3 words
			CAL	D	82	A			Obtain
			COM						the
			ANS	D	83	A			0's row
			CAL	D	82	A			of
			ANA	D	86	A			the
			ORA	D	83	A			card
			SLW	D	82	A			image
			CAL	D	84	A			Obtain
			SLW	D	83	A			x
			CAL	D	85	A			and y
			SLW	D	84	A			rows
			TIX	BCD	1	A		32	Go back for right half image
			LXA	D	25	A			Copy
		COPY	CPY	D	21	A			loop
			CPY	D	53	A			for
			TIX	COPY		A		1	printing
			TRA	D	55				Final exit
		D							Storage location
		D + 1							of six
		D + 2							words of bcd
		D + 3							characters
									for left half
									of card image

FIGURE 52a

Identification	Cards	Location	Operation Code	Sign / Class	Address Symbolic	Address Absolute	Tag	Class	Decrement Symbolic	Decrement Absolute	Comments
		D + 6									Numerical 8-4
											8-3
											8-2
											9 9
											8 8
											7 7
											6 6
											5 5
											4 4
											3 3
											2 2
											1 1
		D + 18									0 0
											zone 11 x
											10 y
											01
		D + 22									00
		D + 23	OCT	-							Loc. of minus zero
		D + 24	OCT		2	000006					Loc. of 2 and 6
		D + 25	FXD			12					Loc. of 12
		D + 26	OCT		21	000100					Loc. of 17 and 64
		D + 27									Column indicator storage
		D + 28	OCT	-	373737	373737					Mask
											Storage for index B
											Unused
											locations
		D + 32									Storage location
											of six
											words of bcd
											characters for
											right half
											of card image
		D + 38									Right half of the
		.....									card image is
		.....									formed and
		.....									stored exactly
		.....									as left half
		D + 55									Main program continues here

FIGURE 52b

Identification	Cards	Location	Operation Code	Sign / Class	Address Symbolic	Address Absolute	Tag	Class	Decrement Symbolic	Decrement Absolute	Comments
		C. SEQ	SXD		CBOX		C				Save contents of C
			TSX		SINX		C				Transfer to SIN X
											Storage for X
			LXD		CBOX		C				Restore contents of C
		-----	-----		----------	-----					
		CBOX									Erasable storage in main program

FIGURE 53

Identification	Cards	Location	Operation Code	Sign / Class	Address Symbolic	Address Absolute	Tag	Class	Decrement Symbolic	Decrement Absolute	Comments
		SINX	CLA			1	C				Place X in AC
			STO		SINX	+ i					Store X
			SXD				C				Save index C
			---		-------	-----					Computation for sin x
		BOXC									Storage for C in subroutine
			---		-------	-----					
			LXD				C				Restore C
			TRA			2	C				Exit to main program

FIGURE 54

APPENDIX A

BINARY AND OCTAL NUMBER SYSTEMS

IN THE FAMILIAR decimal system of representing numbers, a number is expressed by a sum of terms. Each individual term consists of a product of a power of *ten* and some integer in the set 0, 1, . . . , 9. For example,

$$123 = (1 \times 10^2) + (2 \times 10^1) + (3 \times 10^0).$$
$$28.875 = (2 \times 10^1) + (8 \times 10^0) + (8 \times 10^{-1}) + (7 \times 10^{-2}) + (5 \times 10^{-3})$$

Ten is said to be the base of this system because of the role that the powers of *ten* and the integers up to *ten* play in the above expansions.

If two is chosen as the base, numbers are said to be represented in the binary system. For example, the decimal number

$$\begin{aligned} 123 &= 64 + 32 + 16 + 8 + 2 + 1 \\ &= (1 \times 2^6) + (1 \times 2^5) + (1 \times 2^4) \\ &\quad + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) \\ &\quad + (1 \times 2^0) \end{aligned}$$

If only the coefficients of the powers of *two* are written, the binary representation of 123 becomes 1111011. This is exactly how the form of a decimal number is created. Because the binary representation of a number requires only the two digits 0 and 1, the binary system lends itself naturally to the use of core storage, magnetic drums and tapes, trigger circuits, and so on. These devices can be used to greater advantage and efficiency with the binary system than with the decimal.

Observe that the number systems used in this appendix are subject to the familiar commutative, distributive, and associative laws of arithmetic.

Although binary numbers in general have more terms than their decimal counterparts (on an average, about 3.3 times as many), computation in the binary system is quite simple. The rules for adding binary digits are:

$$0 + 0 = 0$$
$$1 + 0 = 1$$
$$1 + 1 = 10 \text{ (0 with 1 carried)}$$

For example, 1111011 may be added to itself as follows:

```
carries:  1111 11
           1111011
         + 1111011
         ---------
          11110110
```

The rules for subtracting binary digits are equally simple:

$$0 - 0 = 0$$
$$1 - 1 = 0$$
$$1 - 0 = 1$$
$$0 - 1 = 1 \text{ (with 1 borrowed)}$$

For example, 1111011 may be subtracted from 11110110 as follows:

```
borrows:  1111 11
          11110110
         - 1111011
         ---------
           1111011
```

The multiplication table for binary digits is given by

$$0 \times 0 = 0$$
$$1 \times 0 = 0$$
$$1 \times 1 = 1$$

The rules for carrying out mulitiplication and division in longhand are entirely similar to those used with the decimal system. For example, the multiplication of 111 by 101 is done as follows:

```
     111
     101
    ----
     111
    000
   111
  ------
  100011
```

The problem of converting a number represented in decimal to the same number as represented in binary is quite simple.

The binary representation of an *integer* can be obtained from its decimal representation through calcu-

lations carried out in either the decimal or binary system. Calculations in the decimal system will be discussed in this case since it is of most interest. The method consists of successive divisions by two as follows:

Consider the number 123, which may be represented as:

$$123 = 2 \times 61 + 1 = 2\ (32 + 16 + 8 + 4 + 1) + 1$$

Thus, dividing 123 by 2 gives the quotient 61 and remainder 1; this remainder is the coefficient of 2^0 in the binary representation of 123. The coefficient of 2^1 in the binary representation of 123 is equal to the remainder obtained from dividing 61 (the previous quotient) by 2.

$$61 = 2 \times 30 + 1 = 2\ (16 + 8 + 4 + 2) + 1$$

Similarly, 30 divided by 2 gives the remainder zero, corresponding to the fact that the coefficient of 2^2 is 0 in the binary representation of 123. The complete calculation for the conversion of 123 to the binary notation is:

$$\begin{aligned} 123 \div 2 &= 61 + \text{remainder of } 1 \\ 61 \div 2 &= 30 + \text{remainder of } 1 \\ 30 \div 2 &= 15 + \text{remainder of } 0 \\ 15 \div 2 &= 7 + \text{remainder of } 1 \\ 7 \div 2 &= 3 + \text{remainder of } 1 \\ 3 \div 2 &= 1 + \text{remainder of } 1 \\ 1 \div 2 &= 0 + \text{remainder of } 1 \end{aligned}$$

The highest power of 2 appearing in the binary representation of 123 is 2^6 and the coefficient of 2^6 is the quotient 1 obtained in the last step above. The binary representation of 123 therefore consists of seven binary digits and is obtained by writing down, in succession from *right* to *left,* the above six remainders in the order in which they were calculated. Thus,

$$(123)_{10} = (1111011)_2$$

Conversely, to convert a binary integer to decimal form through calculation in the binary system, simply divide successively by ten until a quotient of zero is obtained. The remainders, expressed in decimal notation and written in succession from right to left, give the desired decimal representation. The procedure is very similar to the procedure carried out in the decimal system for converting $(1111011)_2 = (123)_{10}$ and is as follows:

$$\begin{aligned} 1111011 \div 1010 &= 1100 + \text{remainder of } 11 \\ 1100 \div 1010 &= 1 + \text{remainder of } 10 \\ 1 \div 1010 &= 0 + \text{remainder of } 1 \end{aligned}$$

The decimal representation then consists of the integers 1, 10, and 11 in that order from left to right. When converted to decimal, these binary integers give 1, 2, and 3, respectively. Consequently the decimal representation of 1111011 is 123.

The binary representation of a proper decimal fraction may be generated digit by digit through successive multiplication by two in the decimal system. The procedure is somewhat like the integer conversion procedure described above.

For example, consider the decimal fraction .625. In order to illustrate the technique, assume that the answer is already known. Thus,

$$\begin{aligned} .625 &= (1 \times 2^{-1}) + (0 \times 2^{-2}) + 1 \times 2^{-3} \\ &= 0.101 \text{ (in binary)} \end{aligned}$$

A multiplication by two gives:

$$1.25 = 1.01 \text{ (in binary)}$$

The integral parts of each number are equal to one and thus represent the first binary bit to the right of the point. Now, using the fractional parts of the two numbers above, we have by a multiplication of two:

$$\begin{aligned} 2 \times .25 &= 0.5 \\ 10 \times .01 &= 0.1 \text{ (in binary)} \end{aligned}$$

The integral parts of the result equal zero; this is the second binary bit to the right of the point.

Similarly, the third binary bit is calculated by

$$\begin{aligned} 2 \times .5 &= 1.0 \\ 10 \times .1 &= 1.0 \end{aligned}$$

and the third binary bit to the right of the point is seen to be 1.

Of course, further multiplications with the remaining fractional parts would result in zeros. So the conversion is complete, and

$$.625 = .101 \text{ (in binary)}$$

Not all terminating decimal fractions, however, can be represented as termining binary fractions. The following calculations follow the pattern as outlined above for converting 0.35 to binary. A position with an x indicates that this position has not yet been determined.

Calculation	Binary Representation
0.35	0.xxxxxxxxxxxxxx . . .
$2 \times 0.35 = 0.7$	0.0xxxxxxxxxxxxx . . .
$2 \times .7 = 1.4$	0.01xxxxxxxxxxxx . . .
$2 \times .4 = 0.8$	0.010xxxxxxxxxxx . . .
$2 \times .8 = 1.6$	0.0101xxxxxxxxxx . . .
$2 \times .6 = 1.2$	0.01011xxxxxxxxx . . .
$2 \times .2 = 0.4$	0.010110xxxxxxxx . . .
$2 \times .4 = 0.8$	0.0101100xxxxxxx . . .
$2 \times .8 = 1.6$	0.01011001xxxxxx . . .
$2 \times .6 = 1.2$	0.010110011xxxxx . . .
$2 \times .2 = 0.4$	0.0101100110xxxx . . .
$2 \times .4 = 0.8$	0.01011001100xxx . . .

It is evident that this procedure will repeat indefinitely. Thus, the terminating decimal fraction 0.35 equals the non-terminating recurring binary fraction. An approximate representation of 0.35 as a terminating binary fraction may be obtained by rounding. For instance, rounding to eight binary places may be done by adding one-half in the eighth place, which in the binary system is equivalent to adding 1 in the ninth place, as follows:

$$\begin{array}{l|l} 0.01011001 & 10011001100\ldots \\ & 1 \\ \hline 0.01011010 & 00011001100\ldots \end{array}$$

The approximate eight-place binary representation of 0.35 is therefore

$$0.01011010$$

Improper decimal fractions may be converted to binary by converting the integral and fractional parts separately or by scaling the number so that it becomes an integer or a proper fraction. For example,

$$(28.875)_{10} = (11100.111)_2$$

Or, to convert the decimal number 15.625, first write it as a proper fraction times a scale factor:

$$15.625 = 0.15625 \times 10^2$$

Then, converting each factor separately gives:

$$\begin{aligned}(15.625)_{10} &= (0.00101)_2 \times (1100100)_2 \\ &= (1111.101)_2\end{aligned}$$

If the base 8 is chosen for representing numbers, the representation is said to be in the octal system. For example:

$$(123)_{10} = 1 \times 8^2 + 7 \times 8^1 + 3 \times 8^0 = (173)_8$$

The decimal-to-octal conversion of an integer can be effected by dividing successively by 8 (in the decimal system) until a quotient of zero is obtained. The octal representation then consists of the successive remainders written in order from right to left.

Octal-to-binary conversion is particularly simple. Because 8 equals 2^3, the conversion is carried out merely by replacing the octal digits with their binary equivalents expressed as three-digit binary numbers. For example, to convert the octal number 173, simply replace 3 with 011, 7 with 111, and 1 with 001, and obtain 1111011, omitting the zeros on the extreme left. Conversely, to pass from binary to octal, simply arrange the binary digits in groups of three, beginning at the binary point and proceeding to the left and to the right. Fill out with zeros at the extreme right or left if necessary. Then replace each group of three binary digits with its octal equivalent. For example:

$$(11100.111)_2 = (011,100.111)_2 = (34.7)_8$$

Thus, the octal notation furnishes a convenient shorthand for the binary notation, especially for handling large numbers. The integer whose binary representation consists of thirty-five 1's is the largest integer that can be represented by thirty-five binary digits. In decimal notation this integer, which equals $2^{35} - 1$, requires eleven digits:

34,359,738,367

Its octal representation requires twelve digits:

377,777,777,777

Hence, in this case the octal system is only a little less economical of notation than the decimal system but is considerably more economical than the binary system.

APPENDIX B

TABLE OF POWERS OF 2

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 161 193 847 656 25
134 217 728	27	0.000 000 007 450 580 596 923 828 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25
68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 806 640 625
137 438 953 472	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944	38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125

APPENDIX C. OCTAL-DECIMAL INTEGER CONVERSION TABLE

0000 to 0777 (Octal) | 0000 to 0511 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
0000	0000	0001	0002	0003	0004	0005	0006	0007
0010	0008	0009	0010	0011	0012	0013	0014	0015
0020	0016	0017	0018	0019	0020	0021	0022	0023
0030	0024	0025	0026	0027	0028	0029	0030	0031
0040	0032	0033	0034	0035	0036	0037	0038	0039
0050	0040	0041	0042	0043	0044	0045	0046	0047
0060	0048	0049	0050	0051	0052	0053	0054	0055
0070	0056	0057	0058	0059	0060	0061	0062	0063
0100	0064	0065	0066	0067	0068	0069	0070	0071
0110	0072	0073	0074	0075	0076	0077	0078	0079
0120	0080	0081	0082	0083	0084	0085	0086	0087
0130	0088	0089	0090	0091	0092	0093	0094	0095
0140	0096	0097	0098	0099	0100	0101	0102	0103
0150	0104	0105	0106	0107	0108	0109	0110	0111
0160	0112	0113	0114	0115	0116	0117	0118	0119
0170	0120	0121	0122	0123	0124	0125	0126	0127
0200	0128	0129	0130	0131	0132	0133	0134	0135
0210	0136	0137	0138	0139	0140	0141	0142	0143
0220	0144	0145	0146	0147	0148	0149	0150	0151
0230	0152	0153	0154	0155	0156	0157	0158	0159
0240	0160	0161	0162	0163	0164	0165	0166	0167
0250	0168	0169	0170	0171	0172	0173	0174	0175
0260	0176	0177	0178	0179	0180	0181	0182	0183
0270	0184	0185	0186	0187	0188	0189	0190	0191
0300	0192	0193	0194	0195	0196	0197	0198	0199
0310	0200	0201	0202	0203	0204	0205	0206	0207
0320	0208	0209	0210	0211	0212	0213	0214	0215
0330	0216	0217	0218	0219	0220	0221	0222	0223
0340	0224	0225	0226	0227	0228	0229	0230	0231
0350	0232	0233	0234	0235	0236	0237	0238	0239
0360	0240	0241	0242	0243	0244	0245	0246	0247
0370	0248	0249	0250	0251	0252	0253	0254	0255

	0	1	2	3	4	5	6	7
0400	0256	0257	0258	0259	0260	0261	0262	0263
0410	0264	0265	0266	0267	0268	0269	0270	0271
0420	0272	0273	0274	0275	0276	0277	0278	0279
0430	0280	0281	0282	0283	0284	0285	0286	0287
0440	0288	0289	0290	0291	0292	0293	0294	0295
0450	0296	0297	0298	0299	0300	0301	0302	0303
0460	0304	0305	0306	0307	0308	0309	0310	0311
0470	0312	0313	0314	0315	0316	0317	0318	0319
0500	0320	0321	0322	0323	0324	0325	0326	0327
0510	0328	0329	0330	0331	0332	0333	0334	0335
0520	0336	0337	0338	0339	0340	0341	0342	0343
0530	0344	0345	0346	0347	0348	0349	0350	0351
0540	0352	0353	0354	0355	0356	0357	0358	0359
0550	0360	0361	0362	0363	0364	0365	0366	0367
0560	0368	0369	0370	0371	0372	0373	0374	0375
0570	0376	0377	0378	0379	0380	0381	0382	0383
0600	0384	0385	0386	0387	0388	0389	0390	0391
0610	0392	0393	0394	0395	0396	0397	0398	0399
0620	0400	0401	0402	0403	0404	0405	0406	0407
0630	0408	0409	0410	0411	0412	0413	0414	0415
0640	0416	0417	0418	0419	0420	0421	0422	0423
0650	0424	0425	0426	0427	0428	0429	0430	0431
0660	0432	0433	0434	0435	0436	0437	0438	0439
0670	0440	0441	0442	0443	0444	0445	0446	0447
0700	0448	0449	0450	0451	0452	0453	0454	0455
0710	0456	0457	0458	0459	0460	0461	0462	0463
0720	0464	0465	0466	0467	0468	0469	0470	0471
0730	0472	0473	0474	0475	0476	0477	0478	0479
0740	0480	0481	0482	0483	0484	0485	0486	0487
0750	0488	0489	0490	0491	0492	0493	0494	0495
0760	0496	0497	0498	0499	0500	0501	0502	0503
0770	0504	0505	0506	0507	0508	0509	0510	0511

1000 to 1777 (Octal) | 0512 to 1023 (Decimal)

	0	1	2	3	4	5	6	7
1000	0512	0513	0514	0515	0516	0517	0518	0519
1010	0520	0521	0522	0523	0524	0525	0526	0527
1020	0528	0529	0530	0531	0532	0533	0534	0535
1030	0536	0537	0538	0539	0540	0541	0542	0543
1040	0544	0545	0546	0547	0548	0549	0550	0551
1050	0552	0553	0554	0555	0556	0557	0558	0559
1060	0560	0561	0562	0563	0564	0565	0566	0567
1070	0568	0569	0570	0571	0572	0573	0574	0575
1100	0576	0577	0578	0579	0580	0581	0582	0583
1110	0584	0585	0586	0587	0588	0589	0590	0591
1120	0592	0593	0594	0595	0596	0597	0598	0599
1130	0600	0601	0602	0603	0604	0605	0606	0607
1140	0608	0609	0610	0611	0612	0613	0614	0615
1150	0616	0617	0618	0619	0620	0621	0622	0623
1160	0624	0625	0626	0627	0628	0629	0630	0631
1170	0632	0633	0634	0635	0636	0637	0638	0639
1200	0640	0641	0642	0643	0644	0645	0646	0647
1210	0648	0649	0650	0651	0652	0653	0654	0655
1220	0656	0657	0658	0659	0660	0661	0662	0663
1230	0664	0665	0666	0667	0668	0669	0670	0671
1240	0672	0673	0674	0675	0676	0677	0678	0679
1250	0680	0681	0682	0683	0684	0685	0686	0687
1260	0688	0689	0690	0691	0692	0693	0694	0695
1270	0696	0697	0698	0699	0700	0701	0702	0703
1300	0704	0705	0706	0707	0708	0709	0710	0711
1310	0712	0713	0714	0715	0716	0717	0718	0719
1320	0720	0721	0722	0723	0724	0725	0726	0727
1330	0728	0729	0730	0731	0732	0733	0734	0735
1340	0736	0737	0738	0739	0740	0741	0742	0743
1350	0744	0745	0746	0747	0748	0749	0750	0751
1360	0752	0753	0754	0755	0756	0757	0758	0759
1370	0760	0761	0762	0763	0764	0765	0766	0767

	0	1	2	3	4	5	6	7
1400	0768	0769	0770	0771	0772	0773	0774	0775
1410	0776	0777	0778	0779	0780	0781	0782	0783
1420	0784	0785	0786	0787	0788	0789	0790	0791
1430	0792	0793	0794	0795	0796	0797	0798	0799
1440	0800	0801	0802	0803	0804	0805	0806	0807
1450	0808	0809	0810	0811	0812	0813	0814	0815
1460	0816	0817	0818	0819	0820	0821	0822	0823
1470	0824	0825	0826	0827	0828	0829	0830	0831
1500	0832	0833	0834	0835	0836	0837	0838	0839
1510	0840	0841	0842	0843	0844	0845	0846	0847
1520	0848	0849	0850	0851	0852	0853	0854	0855
1530	0856	0857	0858	0859	0860	0861	0862	0863
1540	0864	0865	0866	0867	0868	0869	0870	0871
1550	0872	0873	0874	0875	0876	0877	0878	0879
1560	0880	0881	0882	0883	0884	0885	0886	0887
1570	0888	0889	0890	0891	0892	0893	0894	0895
1600	0896	0897	0898	0899	0900	0901	0902	0903
1610	0904	0905	0906	0907	0908	0909	0910	0911
1620	0912	0913	0914	0915	0916	0917	0918	0919
1630	0920	0921	0922	0923	0924	0925	0926	0927
1640	0928	0929	0930	0931	0932	0933	0934	0935
1650	0936	0937	0938	0939	0940	0941	0942	0943
1660	0944	0945	0946	0947	0948	0949	0950	0951
1670	0952	0953	0954	0955	0956	0957	0958	0959
1700	0960	0961	0962	0963	0964	0965	0966	0967
1710	0968	0969	0970	0971	0972	0973	0974	0975
1720	0976	0977	0978	0979	0980	0981	0982	0983
1730	0984	0985	0986	0987	0988	0989	0990	0991
1740	0992	0993	0994	0995	0996	0997	0998	0999
1750	1000	1001	1002	1003	1004	1005	1006	1007
1760	1008	1009	1010	1011	1012	1013	1014	1015
1770	1016	1017	1018	1019	1020	1021	1022	1023

OCTAL-DECIMAL INTEGER CONVERSION TABLE

	0	1	2	3	4	5	6	7
2000	1024	1025	1026	1027	1028	1029	1030	1031
2010	1032	1033	1034	1035	1036	1037	1038	1039
2020	1040	1041	1042	1043	1044	1045	1046	1047
2030	1048	1049	1050	1051	1052	1053	1054	1055
2040	1056	1057	1058	1059	1060	1061	1062	1063
2050	1064	1065	1066	1067	1068	1069	1070	1071
2060	1072	1073	1074	1075	1076	1077	1078	1079
2070	1080	1081	1082	1083	1084	1085	1086	1087
2100	1088	1089	1090	1091	1092	1093	1094	1095
2110	1096	1097	1098	1099	1100	1101	1102	1103
2120	1104	1105	1106	1107	1108	1109	1110	1111
2130	1112	1113	1114	1115	1116	1117	1118	1119
2140	1120	1121	1122	1123	1124	1125	1126	1127
2150	1128	1129	1130	1131	1132	1133	1134	1135
2160	1136	1137	1138	1139	1140	1141	1142	1143
2170	1144	1145	1146	1147	1148	1149	1150	1151
2200	1152	1153	1154	1155	1156	1157	1158	1159
2210	1160	1161	1162	1163	1164	1165	1166	1167
2220	1168	1169	1170	1171	1172	1173	1174	1175
2230	1176	1177	1178	1179	1180	1181	1182	1183
2240	1184	1185	1186	1187	1188	1189	1190	1191
2250	1192	1193	1194	1195	1196	1197	1198	1199
2260	1200	1201	1202	1203	1204	1205	1206	1207
2270	1208	1209	1210	1211	1212	1213	1214	1215
2300	1216	1217	1218	1219	1220	1221	1222	1223
2310	1224	1225	1226	1227	1228	1229	1230	1231
2320	1232	1233	1234	1235	1236	1237	1238	1239
2330	1240	1241	1242	1243	1244	1245	1246	1247
2340	1248	1249	1250	1251	1252	1253	1254	1255
2350	1256	1257	1258	1259	1260	1261	1262	1263
2360	1264	1265	1266	1267	1268	1269	1270	1271
2370	1272	1273	1274	1275	1276	1277	1278	1279

	0	1	2	3	4	5	6	7
2400	1280	1281	1282	1283	1284	1285	1286	1287
2410	1288	1289	1290	1291	1292	1293	1294	1295
2420	1296	1297	1298	1299	1300	1301	1302	1303
2430	1304	1305	1306	1307	1308	1309	1310	1311
2440	1312	1313	1314	1315	1316	1317	1318	1319
2450	1320	1321	1322	1323	1324	1325	1326	1327
2460	1328	1329	1330	1331	1332	1333	1334	1335
2470	1336	1337	1338	1339	1340	1341	1342	1343
2500	1344	1345	1346	1347	1348	1349	1350	1351
2510	1352	1353	1354	1355	1356	1357	1358	1359
2520	1360	1361	1362	1363	1364	1365	1366	1367
2530	1368	1369	1370	1371	1372	1373	1374	1375
2540	1376	1377	1378	1379	1380	1381	1382	1383
2550	1384	1385	1386	1387	1388	1389	1390	1391
2560	1392	1393	1394	1395	1396	1397	1398	1399
2570	1400	1401	1402	1403	1404	1405	1406	1407
2600	1408	1409	1410	1411	1412	1413	1414	1415
2610	1416	1417	1418	1419	1420	1421	1422	1423
2620	1424	1425	1426	1427	1428	1429	1430	1431
2630	1432	1433	1434	1435	1436	1437	1438	1439
2640	1440	1441	1442	1443	1444	1445	1446	1447
2650	1448	1449	1450	1451	1452	1453	1454	1455
2660	1456	1457	1458	1459	1460	1461	1462	1463
2670	1464	1465	1466	1467	1468	1469	1470	1471
2700	1472	1473	1474	1475	1476	1477	1478	1479
2710	1480	1481	1482	1483	1484	1485	1486	1487
2720	1488	1489	1490	1491	1492	1493	1494	1495
2730	1496	1497	1498	1499	1500	1501	1502	1503
2740	1504	1505	1506	1507	1508	1509	1510	1511
2750	1512	1513	1514	1515	1516	1517	1518	1519
2760	1520	1521	1522	1523	1524	1525	1526	1527
2770	1528	1529	1530	1531	1532	1533	1534	1535

Octal	Decimal
2000 to 2777 (Octal)	1024 to 1535 (Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
3000	1536	1537	1538	1539	1540	1541	1542	1543
3010	1544	1545	1546	1547	1548	1549	1550	1551
3020	1552	1553	1554	1555	1556	1557	1558	1559
3030	1560	1561	1562	1563	1564	1565	1566	1567
3040	1568	1569	1570	1571	1572	1573	1574	1575
3050	1576	1577	1578	1579	1580	1581	1582	1583
3060	1584	1585	1586	1587	1588	1589	1590	1591
3070	1592	1593	1594	1595	1596	1597	1598	1599
3100	1600	1601	1602	1603	1604	1605	1606	1607
3110	1608	1609	1610	1611	1612	1613	1614	1615
3120	1616	1617	1618	1619	1620	1621	1622	1623
3130	1624	1625	1626	1627	1628	1629	1630	1631
3140	1632	1633	1634	1635	1636	1637	1638	1639
3150	1640	1641	1642	1643	1644	1645	1646	1647
3160	1648	1649	1650	1651	1652	1653	1654	1655
3170	1656	1657	1658	1659	1660	1661	1662	1663
3200	1664	1665	1666	1667	1668	1669	1670	1671
3210	1672	1673	1674	1675	1676	1677	1678	1679
3220	1680	1681	1682	1683	1684	1685	1686	1687
3230	1688	1689	1690	1691	1692	1693	1694	1695
3240	1696	1697	1698	1699	1700	1701	1702	1703
3250	1704	1705	1706	1707	1708	1709	1710	1711
3260	1712	1713	1714	1715	1716	1717	1718	1719
3270	1720	1721	1722	1723	1724	1725	1726	1727
3300	1728	1729	1730	1731	1732	1733	1734	1735
3310	1736	1737	1738	1739	1740	1741	1742	1743
3320	1744	1745	1746	1747	1748	1749	1750	1751
3330	1752	1753	1754	1755	1756	1757	1758	1759
3340	1760	1761	1762	1763	1764	1765	1766	1767
3350	1768	1769	1770	1771	1772	1773	1774	1775
3360	1776	1777	1778	1779	1780	1781	1782	1783
3370	1784	1785	1786	1787	1788	1789	1790	1791

	0	1	2	3	4	5	6	7
3400	1792	1793	1794	1795	1796	1797	1798	1799
3410	1800	1801	1802	1803	1804	1805	1806	1807
3420	1808	1809	1810	1811	1812	1813	1814	1815
3430	1816	1817	1818	1819	1820	1821	1822	1823
3440	1824	1825	1826	1827	1828	1829	1830	1831
3450	1832	1833	1834	1835	1836	1837	1838	1839
3460	1840	1841	1842	1843	1844	1845	1846	1847
3470	1848	1849	1850	1851	1852	1853	1854	1855
3500	1856	1857	1858	1859	1860	1861	1862	1863
3510	1864	1865	1866	1867	1868	1869	1870	1871
3520	1872	1873	1874	1875	1876	1877	1878	1879
3530	1880	1881	1882	1883	1884	1885	1886	1887
3540	1888	1889	1890	1891	1892	1893	1894	1895
3550	1896	1897	1898	1899	1900	1901	1902	1903
3560	1904	1905	1906	1907	1908	1909	1910	1911
3570	1912	1913	1914	1915	1916	1917	1918	1919
3600	1920	1921	1922	1923	1924	1925	1926	1927
3610	1928	1929	1930	1931	1932	1933	1934	1935
3620	1936	1937	1938	1939	1940	1941	1942	1943
3630	1944	1945	1946	1947	1948	1949	1950	1951
3640	1952	1953	1954	1955	1956	1957	1958	1959
3650	1960	1961	1962	1963	1964	1965	1966	1967
3660	1968	1969	1970	1971	1972	1973	1974	1975
3670	1976	1977	1978	1979	1980	1981	1982	1983
3700	1984	1985	1986	1987	1988	1989	1990	1991
3710	1992	1993	1994	1995	1996	1997	1998	1999
3720	2000	2001	2002	2003	2004	2005	2006	2007
3730	2008	2009	2010	2011	2012	2013	2014	2015
3740	2016	2017	2018	2019	2020	2021	2022	2023
3750	2024	2025	2026	2027	2028	2029	2030	2031
3760	2032	2033	2034	2035	2036	2037	2038	2039
3770	2040	2041	2042	2043	2044	2045	2046	2047

Octal	Decimal
3000 to 3777 (Octal)	1536 to 2047 (Decimal)

4000	2048
to	to
4777	2559
(Octal)	(Decimal)

Octal	Decimal
10000 -	4096
20000 -	8192
30000 -	12288
40000 -	16384
50000 -	20480
60000 -	24576
70000 -	28672

	0	1	2	3	4	5	6	7
4000	2048	2049	2050	2051	2052	2053	2054	2055
4010	2056	2057	2058	2059	2060	2061	2062	2063
4020	2064	2065	2066	2067	2068	2069	2070	2071
4030	2072	2073	2074	2075	2076	2077	2078	2079
4040	2080	2081	2082	2083	2084	2085	2086	2087
4050	2088	2089	2090	2091	2092	2093	2094	2095
4060	2096	2097	2098	2099	2100	2101	2102	2103
4070	2104	2105	2106	2107	2108	2109	2110	2111
4100	2112	2113	2114	2115	2116	2117	2118	2119
4110	2120	2121	2122	2123	2124	2125	2126	2127
4120	2128	2129	2130	2131	2132	2133	2134	2135
4130	2136	2137	2138	2139	2140	2141	2142	2143
4140	2144	2145	2146	2147	2148	2149	2150	2151
4150	2152	2153	2154	2155	2156	2157	2158	2159
4160	2160	2161	2162	2163	2164	2165	2166	2167
4170	2168	2169	2170	2171	2172	2173	2174	2175
4200	2176	2177	2178	2179	2180	2181	2182	2183
4210	2184	2185	2186	2187	2188	2189	2190	2191
4220	2192	2193	2194	2195	2196	2197	2198	2199
4230	2200	2201	2202	2203	2204	2205	2206	2207
4240	2208	2209	2210	2211	2212	2213	2214	2215
4250	2216	2217	2218	2219	2220	2221	2222	2223
4260	2224	2225	2226	2227	2228	2229	2230	2231
4270	2232	2233	2234	2235	2236	2237	2238	2239
4300	2240	2241	2242	2243	2244	2245	2246	2247
4310	2248	2249	2250	2251	2252	2253	2254	2255
4320	2256	2257	2258	2259	2260	2261	2262	2263
4330	2264	2265	2266	2267	2268	2269	2270	2271
4340	2272	2273	2274	2275	2276	2277	2278	2279
4350	2280	2281	2282	2283	2284	2285	2286	2287
4360	2288	2289	2290	2291	2292	2293	2294	2295
4370	2296	2297	2298	2299	2300	2301	2302	2303

	0	1	2	3	4	5	6	7
4400	2304	2305	2306	2307	2308	2309	2310	2311
4410	2312	2313	2314	2315	2316	2317	2318	2319
4420	2320	2321	2322	2323	2324	2325	2326	2327
4430	2328	2329	2330	2331	2332	2333	2334	2335
4440	2336	2337	2338	2339	2340	2341	2342	2343
4450	2344	2345	2346	2347	2348	2349	2350	2351
4460	2352	2353	2354	2355	2356	2357	2358	2359
4470	2360	2361	2362	2363	2364	2365	2366	2367
4500	2368	2369	2370	2371	2372	2373	2374	2375
4510	2376	2377	2378	2379	2380	2381	2382	2383
4520	2384	2385	2386	2387	2388	2389	2390	2391
4530	2392	2393	2394	2395	2396	2397	2398	2399
4540	2400	2401	2402	2403	2404	2405	2406	2407
4550	2408	2409	2410	2411	2412	2413	2414	2415
4560	2416	2417	2418	2419	2420	2421	2422	2423
4570	2424	2425	2426	2427	2428	2429	2430	2431
4600	2432	2433	2434	2435	2436	2437	2438	2439
4610	2440	2441	2442	2443	2444	2445	2446	2447
4620	2448	2449	2450	2451	2452	2453	2454	2455
4630	2456	2457	2458	2459	2460	2461	2462	2463
4640	2464	2465	2466	2467	2468	2469	2470	2471
4650	2472	2473	2474	2475	2476	2477	2478	2479
4660	2480	2481	2482	2483	2484	2485	2486	2487
4670	2488	2489	2490	2491	2492	2493	2494	2495
4700	2496	2497	2498	2499	2500	2501	2502	2503
4710	2504	2505	2506	2507	2508	2509	2510	2511
4720	2512	2513	2514	2515	2516	2517	2518	2519
4730	2520	2521	2522	2523	2524	2525	2526	2527
4740	2528	2529	2530	2531	2532	2533	2534	2535
4750	2536	2537	2538	2539	2540	2541	2542	2543
4760	2544	2545	2546	2547	2548	2549	2550	2551
4770	2552	2553	2554	2555	2556	2557	2558	2559

5000	2560
to	to
5777	3071
(Octal)	(Decimal)

	0	1	2	3	4	5	6	7
5000	2560	2561	2562	2563	2564	2565	2566	2567
5010	2568	2569	2570	2571	2572	2573	2574	2575
5020	2576	2577	2578	2579	2580	2581	2582	2583
5030	2584	2585	2586	2587	2588	2589	2590	2591
5040	2592	2593	2594	2595	2596	2597	2598	2599
5050	2600	2601	2602	2603	2604	2605	2606	2607
5060	2608	2609	2610	2611	2612	2613	2614	2615
5070	2616	2617	2618	2619	2620	2621	2622	2623
5100	2624	2625	2626	2627	2628	2629	2630	2631
5110	2632	2633	2634	2635	2636	2637	2638	2639
5120	2640	2641	2642	2643	2644	2645	2646	2647
5130	2648	2649	2650	2651	2652	2653	2654	2655
5140	2656	2657	2658	2659	2660	2661	2662	2663
5150	2664	2665	2666	2667	2668	2669	2670	2671
5160	2672	2673	2674	2675	2676	2677	2678	2679
5170	2680	2681	2682	2683	2684	2685	2686	2687
5200	2688	2689	2690	2691	2692	2693	2694	2695
5210	2696	2697	2698	2699	2700	2701	2702	2703
5220	2704	2705	2706	2707	2708	2709	2710	2711
5230	2712	2713	2714	2715	2716	2717	2718	2719
5240	2720	2721	2722	2723	2724	2725	2726	2727
5250	2728	2729	2730	2731	2732	2733	2734	2735
5260	2736	2737	2738	2739	2740	2741	2742	2743
5270	2744	2745	2746	2747	2748	2749	2750	2751
5300	2752	2753	2754	2755	2756	2757	2758	2759
5310	2760	2761	2762	2763	2764	2765	2766	2767
5320	2768	2769	2770	2771	2772	2773	2774	2775
5330	2776	2777	2778	2779	2780	2781	2782	2783
5340	2784	2785	2786	2787	2788	2789	2790	2791
5350	2792	2793	2794	2795	2796	2797	2798	2799
5360	2800	2801	2802	2803	2804	2805	2806	2807
5370	2808	2809	2810	2811	2812	2813	2814	2815

	0	1	2	3	4	5	6	7
5400	2816	2817	2818	2819	2820	2821	2822	2823
5410	2824	2825	2826	2827	2828	2829	2830	2831
5420	2832	2833	2834	2835	2836	2837	2838	2839
5430	2840	2841	2842	2843	2844	2845	2846	2847
5440	2848	2849	2850	2851	2852	2853	2854	2855
5450	2856	2857	2858	2859	2860	2861	2862	2863
5460	2864	2865	2866	2867	2868	2869	2870	2871
5470	2872	2873	2874	2875	2876	2877	2878	2879
5500	2880	2881	2882	2883	2884	2885	2886	2887
5510	2888	2889	2890	2891	2892	2893	2894	2895
5520	2896	2897	2898	2899	2900	2901	2902	2903
5530	2904	2905	2906	2907	2908	2909	2910	2911
5540	2912	2913	2914	2915	2916	2917	2918	2919
5550	2920	2921	2922	2923	2924	2925	2926	2927
5560	2928	2929	2930	2931	2932	2933	2934	2935
5570	2936	2937	2938	2939	2940	2941	2942	2943
5600	2944	2945	2946	2947	2948	2949	2950	2951
5610	2952	2953	2954	2955	2956	2957	2958	2959
5620	2960	2961	2962	2963	2964	2965	2966	2967
5630	2968	2969	2970	2971	2972	2973	2974	2975
5640	2976	2977	2978	2979	2980	2981	2982	2983
5650	2984	2985	2986	2987	2988	2989	2990	2991
5660	2992	2993	2994	2995	2996	2997	2998	2999
5670	3000	3001	3002	3003	3004	3005	3006	3007
5700	3008	3009	3010	3011	3012	3013	3014	3015
5710	3016	3017	3018	3019	3020	3021	3022	3023
5720	3024	3025	3026	3027	3028	3029	3030	3031
5730	3032	3033	3034	3035	3036	3037	3038	3039
5740	3040	3041	3042	3043	3044	3045	3046	3047
5750	3048	3049	3050	3051	3052	3053	3054	3055
5760	3056	3057	3058	3059	3060	3061	3062	3063
5770	3064	3065	3066	3067	3068	3069	3070	3071

OCTAL-DECIMAL INTEGER CONVERSION TABLE

	0	1	2	3	4	5	6	7
6000	3072	3073	3074	3075	3076	3077	3078	3079
6010	3080	3081	3082	3083	3084	3085	3086	3087
6020	3088	3089	3090	3091	3092	3093	3094	3095
6030	3096	3097	3098	3099	3100	3101	3102	3103
6040	3104	3105	3106	3107	3108	3109	3110	3111
6050	3112	3113	3114	3115	3116	3117	3118	3119
6060	3120	3121	3122	3123	3124	3125	3126	3127
6070	3128	3129	3130	3131	3132	3133	3134	3135
6100	3136	3137	3138	3139	3140	3141	3142	3143
6110	3144	3145	3146	3147	3148	3149	3150	3151
6120	3152	3153	3154	3155	3156	3157	3158	3159
6130	3160	3161	3162	3163	3164	3165	3166	3167
6140	3168	3169	3170	3171	3172	3173	3174	3175
6150	3176	3177	3178	3179	3180	3181	3182	3183
6160	3184	3185	3186	3187	3188	3189	3190	3191
6170	3192	3193	3194	3195	3196	3197	3198	3199
6200	3200	3201	3202	3203	3204	3205	3206	3207
6210	3208	3209	3210	3211	3212	3213	3214	3215
6220	3216	3217	3218	3219	3220	3221	3222	3223
6230	3224	3225	3226	3227	3228	3229	3230	3231
6240	3232	3233	3234	3235	3236	3237	3238	3239
6250	3240	3241	3242	3243	3244	3245	3246	3247
6260	3248	3249	3250	3251	3252	3253	3254	3255
6270	3256	3257	3258	3259	3260	3261	3262	3263
6300	3264	3265	3266	3267	3268	3269	3270	3271
6310	3272	3273	3274	3275	3276	3277	3278	3279
6320	3280	3281	3282	3283	3284	3285	3286	3287
6330	3288	3289	3290	3291	3292	3293	3294	3295
6340	3296	3297	3298	3299	3300	3301	3302	3303
6350	3304	3305	3306	3307	3308	3309	3310	3311
6360	3312	3313	3314	3315	3316	3317	3318	3319
6370	3320	3321	3322	3323	3324	3325	3326	3327

	0	1	2	3	4	5	6	7
6400	3328	3329	3330	3331	3332	3333	3334	3335
6410	3336	3337	3338	3339	3340	3341	3342	3343
6420	3344	3345	3346	3347	3348	3349	3350	3351
6430	3352	3353	3354	3355	3356	3357	3358	3359
6440	3360	3361	3362	3363	3364	3365	3366	3367
6450	3368	3369	3370	3371	3372	3373	3374	3375
6460	3376	3377	3378	3379	3380	3381	3382	3383
6470	3384	3385	3386	3387	3388	3389	3390	3391
6500	3392	3393	3394	3395	3396	3397	3398	3399
6510	3400	3401	3402	3403	3404	3405	3406	3407
6520	3408	3409	3410	3411	3412	3413	3414	3415
6530	3416	3417	3418	3419	3420	3421	3422	3423
6540	3424	3425	3426	3427	3428	3429	3430	3431
6550	3432	3433	3434	3435	3436	3437	3438	3439
6560	3440	3441	3442	3443	3444	3445	3446	3447
6570	3448	3449	3450	3451	3452	3453	3454	3455
6600	3456	3457	3458	3459	3460	3461	3462	3463
6610	3464	3465	3466	3467	3468	3469	3470	3471
6620	3472	3473	3474	3475	3476	3477	3478	3479
6630	3480	3481	3482	3483	3484	3485	3486	3487
6640	3488	3489	3490	3491	3492	3493	3494	3495
6650	3496	3497	3498	3499	3500	3501	3502	3503
6660	3504	3505	3506	3507	3508	3509	3510	3511
6670	3512	3513	3514	3515	3516	3517	3518	3519
6700	3520	3521	3522	3523	3524	3525	3526	3527
6710	3528	3529	3530	3531	3532	3533	3534	3535
6720	3536	3537	3538	3539	3540	3541	3542	3543
6730	3544	3545	3546	3547	3548	3549	3550	3551
6740	3552	3553	3554	3555	3556	3557	3558	3559
6750	3560	3561	3562	3563	3564	3565	3566	3567
6760	3568	3569	3570	3571	3572	3573	3574	3575
6770	3576	3577	3578	3579	3580	3581	3582	3583

6000 to 6777 (Octal)	3072 to 3583 (Decimal)

Octal	Decimal
10000	4096
20000	8192
30000	12288
40000	16384
50000	20480
60000	24576
70000	28672

	0	1	2	3	4	5	6	7
7000	3584	3585	3586	3587	3588	3589	3590	3591
7010	3592	3593	3594	3595	3596	3597	3598	3599
7020	3600	3601	3602	3603	3604	3605	3606	3607
7030	3608	3609	3610	3611	3612	3613	3614	3615
7040	3616	3617	3618	3619	3620	3621	3622	3623
7050	3624	3625	3626	3627	3628	3629	3630	3631
7060	3632	3633	3634	3635	3636	3637	3638	3639
7070	3640	3641	3642	3643	3644	3645	3646	3647
7100	3648	3649	3650	3651	3652	3653	3654	3655
7110	3656	3657	3658	3659	3660	3661	3662	3663
7120	3664	3665	3666	3667	3668	3669	3670	3671
7130	3672	3673	3674	3675	3676	3677	3678	3679
7140	3680	3681	3682	3683	3684	3685	3686	3687
7150	3688	3689	3690	3691	3692	3693	3694	3695
7160	3696	3697	3698	3699	3700	3701	3702	3703
7170	3704	3705	3706	3707	3708	3709	3710	3711
7200	3712	3713	3714	3715	3716	3717	3718	3719
7210	3720	3721	3722	3723	3724	3725	3726	3727
7220	3728	3729	3730	3731	3732	3733	3734	3735
7230	3736	3737	3738	3739	3740	3741	3742	3743
7240	3744	3745	3746	3747	3748	3749	3750	3751
7250	3752	3753	3754	3755	3756	3757	3758	3759
7260	3760	3761	3762	3763	3764	3765	3766	3767
7270	3768	3769	3770	3771	3772	3773	3774	3775
7300	3776	3777	3778	3779	3780	3781	3782	3783
7310	3784	3785	3786	3787	3788	3789	3790	3791
7320	3792	3793	3794	3795	3796	3797	3798	3799
7330	3800	3801	3802	3803	3804	3805	3806	3807
7340	3808	3809	3810	3811	3812	3813	3814	3815
7350	3816	3817	3818	3819	3820	3821	3822	3823
7360	3824	3825	3826	3827	3828	3829	3830	3831
7370	3832	3833	3834	3835	3836	3837	3838	3839

	0	1	2	3	4	5	6	7
7400	3840	3841	3842	3843	3844	3845	3846	3847
7410	3848	3849	3850	3851	3852	3853	3854	3855
7420	3856	3857	3858	3859	3860	3861	3862	3863
7430	3864	3865	3866	3867	3868	3869	3870	3871
7440	3872	3873	3874	3875	3876	3877	3878	3879
7450	3880	3881	3882	3883	3884	3885	3886	3887
7460	3888	3889	3890	3891	3892	3893	3894	3895
7470	3896	3897	3898	3899	3900	3901	3902	3903
7500	3904	3905	3906	3907	3908	3909	3910	3911
7510	3912	3913	3914	3915	3916	3917	3918	3919
7520	3920	3921	3922	3923	3924	3925	3926	3927
7530	3928	3929	3930	3931	3932	3933	3934	3935
7540	3936	3937	3938	3939	3940	3941	3942	3943
7550	3944	3945	3946	3947	3948	3949	3950	3951
7560	3952	3953	3954	3955	3956	3957	3958	3959
7570	3960	3961	3962	3963	3964	3965	3966	3967
7600	3968	3969	3970	3971	3972	3973	3974	3975
7610	3976	3977	3978	3979	3980	3981	3982	3983
7620	3984	3985	3986	3987	3988	3989	3990	3991
7630	3992	3993	3994	3995	3996	3997	3998	3999
7640	4000	4001	4002	4003	4004	4005	4006	4007
7650	4008	4009	4010	4011	4012	4013	4014	4015
7660	4016	4017	4018	4019	4020	4021	4022	4023
7670	4024	4025	4026	4027	4028	4029	4030	4031
7700	4032	4033	4034	4035	4036	4037	4038	4039
7710	4040	4041	4042	4043	4044	4045	4046	4047
7720	4048	4049	4050	4051	4052	4053	4054	4055
7730	4056	4057	4058	4059	4060	4061	4062	4063
7740	4064	4065	4066	4067	4068	4069	4070	4071
7750	4072	4073	4074	4075	4076	4077	4078	4079
7760	4080	4081	4082	4083	4084	4085	4086	4087
7770	4088	4089	4090	4091	4092	4093	4094	4095

7000 to 7777 (Octal)	3584 to 4095 (Decimal)

APPENDIX D. OCTAL-DECIMAL FRACTION CONVERSION TABLE

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000	.000000	.100	.125000	.200	.250000	.300	.375000
.001	.001953	.101	.126953	.201	.251953	.301	.376953
.002	.003906	.102	.128906	.202	.253906	.302	.378906
.003	.005859	.103	.130859	.203	.255859	.303	.380859
.004	.007812	.104	.132812	.204	.257812	.304	.382812
.005	.009765	.105	.134765	.205	.259765	.305	.384765
.006	.011718	.106	.136718	.206	.261718	.306	.386718
.007	.013671	.107	.138671	.207	.263671	.307	.388671
.010	.015625	.110	.140625	.210	.265625	.310	.390625
.011	.017578	.111	.142578	.211	.267578	.311	.392578
.012	.019531	.112	.144531	.212	.269531	.312	.394531
.013	.021484	.113	.146484	.213	.271484	.313	.396484
.014	.023437	.114	.148437	.214	.273437	.314	.398437
.015	.025390	.115	.150390	.215	.275390	.315	.400390
.016	.027343	.116	.152343	.216	.277343	.316	.402343
.017	.029296	.117	.154296	.217	.279296	.317	.404296
.020	.031250	.120	.156250	.220	.281250	.320	.406250
.021	.033203	.121	.158203	.221	.283203	.321	.408203
.022	.035156	.122	.160156	.222	.285156	.322	.410156
.023	.037109	.123	.162109	.223	.287109	.323	.412109
.024	.039062	.124	.164062	.224	.289062	.324	.414062
.025	.041015	.125	.166015	.225	.291015	.325	.416015
.026	.042968	.126	.167968	.226	.292968	.326	.417968
.027	.044921	.127	.169921	.227	.294921	.327	.419921
.030	.046875	.130	.171875	.230	.296875	.330	.421875
.031	.048828	.131	.173828	.231	.298828	.331	.423828
.032	.050781	.132	.175781	.232	.300781	.332	.425781
.033	.052734	.133	.177734	.233	.302734	.333	.427734
.034	.054687	.134	.179687	.234	.304687	.334	.429687
.035	.056640	.135	.181640	.235	.306640	.335	.431640
.036	.058593	.136	.183593	.236	.308593	.336	.433593
.037	.060546	.137	.185546	.237	.310546	.337	.435546
.040	.062500	.140	.187500	.240	.312500	.340	.437500
.041	.064453	.141	.189453	.241	.314453	.341	.439453
.042	.066406	.142	.191406	.242	.316406	.342	.441406
.043	.068359	.143	.193359	.243	.318359	.343	.443359
.044	.070312	.144	.195312	.244	.320312	.344	.445312
.045	.072265	.145	.197265	.245	.322265	.345	.447265
.046	.074218	.146	.199218	.246	.324218	.346	.449218
.047	.076171	.147	.201171	.247	.326171	.347	.451171
.050	.078125	.150	.203125	.250	.328125	.350	.453125
.051	.080078	.151	.205078	.251	.330078	.351	.455078
.052	.082031	.152	.207031	.252	.332031	.352	.457031
.053	.083984	.153	.208984	.253	.333984	.353	.458984
.054	.085937	.154	.210937	.254	.335937	.354	.460937
.055	.087890	.155	.212890	.255	.337890	.355	.462890
.056	.089843	.156	.214843	.256	.339843	.356	.464843
.057	.091796	.157	.216796	.257	.341796	.357	.466796
.060	.093750	.160	.218750	.260	.343750	.360	.468750
.061	.095703	.161	.220703	.261	.345703	.361	.470703
.062	.097656	.162	.222656	.262	.347656	.362	.472656
.063	.099609	.163	.224609	.263	.349609	.363	.474609
.064	.101562	.164	.226562	.264	.351562	.364	.476562
.065	.103515	.165	.228515	.265	.353515	.365	.478515
.066	.105468	.166	.230468	.266	.355468	.366	.480468
.067	.107421	.167	.232421	.267	.357421	.367	.482421
.070	.109375	.170	.234375	.270	.359375	.370	.484375
.071	.111328	.171	.236328	.271	.361328	.371	.486328
.072	.113281	.172	.238281	.272	.363281	.372	.488281
.073	.115234	.173	.240234	.273	.365234	.373	.490234
.074	.117187	.174	.242187	.274	.367187	.374	.492187
.075	.119140	.175	.244140	.275	.369140	.375	.494140
.076	.121093	.176	.246093	.276	.371093	.376	.496093
.077	.123046	.177	.248046	.277	.373046	.377	.498046

OCTAL-DECIMAL FRACTION CONVERSION TABLE

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000000	.000000	.000100	.000244	.000200	.000488	.000300	.000732
.000001	.000003	.000101	.000247	.000201	.000492	.000301	.000736
.000002	.000007	.000102	.000251	.000202	.000495	.000302	.000740
.000003	.000011	.000103	.000255	.000203	.000499	.000303	.000743
.000004	.000015	.000104	.000259	.000204	.000503	.000304	.000747
.000005	.000019	.000105	.000263	.000205	.000507	.000305	.000751
.000006	.000022	.000106	.000267	.000206	.000511	.000306	.000755
.000007	.000026	.000107	.000270	.000207	.000514	.000307	.000759
.000010	.000030	.000110	.000274	.000210	.000518	.000310	.000762
.000011	.000034	.000111	.000278	.000211	.000522	.000311	.000766
.000012	.000038	.000112	.000282	.000212	.000526	.000312	.000770
.000013	.000041	.000113	.000286	.000213	.000530	.000313	.000774
.000014	.000045	.000114	.000289	.000214	.000534	.000314	.000778
.000015	.000049	.000115	.000293	.000215	.000537	.000315	.000782
.000016	.000053	.000116	.000297	.000216	.000541	.000316	.000785
.000017	.000057	.000117	.000301	.000217	.000545	.000317	.000789
.000020	.000061	.000120	.000305	.000220	.000549	.000320	.000793
.000021	.000064	.000121	.000308	.000221	.000553	.000321	.000797
.000022	.000068	.000122	.000312	.000222	.000556	.000322	.000801
.000023	.000072	.000123	.000316	.000223	.000560	.000323	.000805
.000024	.000076	.000124	.000320	.000224	.000564	.000324	.000808
.000025	.000080	.000125	.000324	.000225	.000568	.000325	.000812
.000026	.000083	.000126	.000328	.000226	.000572	.000326	.000816
.000027	.000087	.000127	.000331	.000227	.000576	.000327	.000820
.000030	.000091	.000130	.000335	.000230	.000579	.000330	.000823
.000031	.000095	.000131	.000339	.000231	.000583	.000331	.000827
.000032	.000099	.000132	.000343	.000232	.000587	.000332	.000831
.000033	.000102	.000133	.000347	.000233	.000591	.000333	.000835
.000034	.000106	.000134	.000350	.000234	.000595	.000334	.000839
.000035	.000110	.000135	.000354	.000235	.000598	.000335	.000843
.000036	.000114	.000136	.000358	.000236	.000602	.000336	.000846
.000037	.000118	.000137	.000362	.000237	.000606	.000337	.000850
.000040	.000122	.000140	.000366	.000240	.000610	.000340	.000854
.000041	.000125	.000141	.000370	.000241	.000614	.000341	.000858
.000042	.000129	.000142	.000373	.000242	.000617	.000342	.000862
.000043	.000133	.000143	.000377	.000243	.000621	.000343	.000865
.000044	.000137	.000144	.000381	.000244	.000625	.000344	.000869
.000045	.000141	.000145	.000385	.000245	.000629	.000345	.000873
.000046	.000144	.000146	.000389	.000246	.000633	.000346	.000877
.000047	.000148	.000147	.000392	.000247	.000637	.000347	.000881
.000050	.000152	.000150	.000396	.000250	.000640	.000350	.000885
.000051	.000156	.000151	.000400	.000251	.000644	.000351	.000888
.000052	.000160	.000152	.000404	.000252	.000648	.000352	.000892
.000053	.000164	.000153	.000408	.000253	.000652	.000353	.000896
.000054	.000167	.000154	.000411	.000254	.000656	.000354	.000900
.000055	.000171	.000155	.000415	.000255	.000659	.000355	.000904
.000056	.000175	.000156	.000419	.000256	.000663	.000356	.000907
.000057	.000179	.000157	.000423	.000257	.000667	.000357	.000911
.000060	.000183	.000160	.000427	.000260	.000671	.000360	.000915
.000061	.000186	.000161	.000431	.000261	.000675	.000361	.000919
.000062	.000190	.000162	.000434	.000262	.000679	.000362	.000923
.000063	.000194	.000163	.000438	.000263	.000682	.000363	.000926
.000064	.000198	.000164	.000442	.000264	.000686	.000364	.000930
.000065	.000202	.000165	.000446	.000265	.000690	.000365	.000934
.000066	.000205	.000166	.000450	.000266	.000694	.000366	.000938
.000067	.000209	.000167	.000453	.000267	.000698	.000367	.000942
.000070	.000213	.000170	.000457	.000270	.000701	.000370	.000946
.000071	.000217	.000171	.000461	.000271	.000705	.000371	.000949
.000072	.000221	.000172	.000465	.000272	.000709	.000372	.000953
.000073	.000225	.000173	.000469	.000273	.000713	.000373	.000957
.000074	.000228	.000174	.000473	.000274	.000717	.000374	.000961
.000075	.000232	.000175	.000476	.000275	.000720	.000375	.000965
.000076	.000236	.000176	.000480	.000276	.000724	.000376	.000968
.000077	.000240	.000177	.000484	.000277	.000728	.000377	.000972

OCTAL-DECIMAL FRACTION CONVERSION TABLE

OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.	OCTAL	DEC.
.000400	.000976	.000500	.001220	.000600	.001464	.000700	.001708
.000401	.000980	.000501	.001224	.000601	.001468	.000701	.001712
.000402	.000984	.000502	.001228	.000602	.001472	.000702	.001716
.000403	.000988	.000503	.001232	.000603	.001476	.000703	.001720
.000404	.000991	.000504	.001235	.000604	.001480	.000704	.001724
.000405	.000995	.000505	.001239	.000605	.001483	.000705	.001728
.000406	.000999	.000506	.001243	.000606	.001487	.000706	.001731
.000407	.001003	.000507	.001247	.000607	.001491	.000707	.001735
.000410	.001007	.000510	.001251	.000610	.001495	.000710	.001739
.000411	.001010	.000511	.001255	.000611	.001499	.000711	.001743
.000412	.001014	.000512	.001258	.000612	.001502	.000712	.001747
.000413	.001018	.000513	.001262	.000613	.001506	.000713	.001750
.000414	.001022	.000514	.001266	.000614	.001510	.000714	.001754
.000415	.001026	.000515	.001270	.000615	.001514	.000715	.001758
.000416	.001029	.000516	.001274	.000616	.001518	.000716	.001762
.000417	.001033	.000517	.001277	.000617	.001522	.000717	.001766
.000420	.001037	.000520	.001281	.000620	.001525	.000720	.001770
.000421	.001041	.000521	.001285	.000621	.001529	.000721	.001773
.000422	.001045	.000522	.001289	.000622	.001533	.000722	.001777
.000423	.001049	.000523	.001293	.000623	.001537	.000723	.001781
.000424	.001052	.000524	.001296	.000624	.001541	.000724	.001785
.000425	.001056	.000525	.001300	.000625	.001544	.000725	.001789
.000426	.001060	.000526	.001304	.000626	.001548	.000726	.001792
.000427	.001064	.000527	.001308	.000627	.001552	.000727	.001796
.000430	.001068	.000530	.001312	.000630	.001556	.000730	.001800
.000431	.001071	.000531	.001316	.000631	.001560	.000731	.001804
.000432	.001075	.000532	.001319	.000632	.001564	.000732	.001808
.000433	.001079	.000533	.001323	.000633	.001567	.000733	.001811
.000434	.001083	.000534	.001327	.000634	.001571	.000734	.001815
.000435	.001087	.000535	.001331	.000635	.001575	.000735	.001819
.000436	.001091	.000536	.001335	.000636	.001579	.000736	.001823
.000437	.001094	.000537	.001338	.000637	.001583	.000737	.001827
.000440	.001098	.000540	.001342	.000640	.001586	.000740	.001831
.000441	.001102	.000541	.001346	.000641	.001590	.000741	.001834
.000442	.001106	.000542	.001350	.000642	.001594	.000742	.001838
.000443	.001110	.000543	.001354	.000643	.001598	.000743	.001842
.000444	.001113	.000544	.001358	.000644	.001602	.000744	.001846
.000445	.001117	.000545	.001361	.000645	.001605	.000745	.001850
.000446	.001121	.000546	.001365	.000646	.001609	.000746	.001853
.000447	.001125	.000547	.001369	.000647	.001613	.000747	.001857
.000450	.001129	.000550	.001373	.000650	.001617	.000750	.001861
.000451	.001132	.000551	.001377	.000651	.001621	.000751	.001865
.000452	.001136	.000552	.001380	.000652	.001625	.000752	.001869
.000453	.001140	.000553	.001384	.000653	.001628	.000753	.001873
.000454	.001144	.000554	.001388	.000654	.001632	.000754	.001876
.000455	.001148	.000555	.001392	.000655	.001636	.000755	.001880
.000456	.001152	.000556	.001396	.000656	.001640	.000756	.001884
.000457	.001155	.000557	.001399	.000657	.001644	.000757	.001888
.000460	.001159	.000560	.001403	.000660	.001647	.000760	.001892
.000461	.001163	.000561	.001407	.000661	.001651	.000761	.001895
.000462	.001167	.000562	.001411	.000662	.001655	.000762	.001899
.000463	.001171	.000563	.001415	.000663	.001659	.000763	.001903
.000464	.001174	.000564	.001419	.000664	.001663	.000764	.001907
.000465	.001178	.000565	.001422	.000665	.001667	.000765	.001911
.000466	.001182	.000566	.001426	.000666	.001670	.000766	.001914
.000467	.001186	.000567	.001430	.000667	.001674	.000767	.001918
.000470	.001190	.000570	.001434	.000670	.001678	.000770	.001922
.000471	.001194	.000571	.001438	.000671	.001682	.000771	.001926
.000472	.001197	.000572	.001441	.000672	.001686	.000772	.001930
.000473	.001201	.000573	.001445	.000673	.001689	.000773	.001934
.000474	.001205	.000574	.001449	.000674	.001693	.000774	.001937
.000475	.001209	.000575	.001453	.000675	.001697	.000775	.001941
.000476	.001213	.000576	.001457	.000676	.001701	.000776	.001945
.000477	.001216	.000577	.001461	.000677	.001705	.000777	.001949

APPENDIX E. OPERATIONS BY ALPHABETIC CODE

Alpha Code	Octal Code	Cycles	Operation	Page
ACL	+ 0361	2	Add and Carry Logical Word	19
ADD	+ 0400	2	Add	17
ADM	+ 0401	2	Add Magnitude	18
ALS	+ 0767	2 I	Accumulator Left Shift	20
ANA	— 0320	3	AND to Accumulator	19
ANS	+ 0320	4	AND to Storage	20
ARS	+ 0771	2 I	Accumulator Right Shift	20
BST	+ 0764	2 III	Backspace Tape	26
CAL	— 0500	2	Clear and Add Logical Word	19
CAS	+ 0340	3	Compare Accumulator with Storage	25
CHS	+ 0760. . .002	2	Change Sign	19
CLA	+ 0500	2	Clear and Add	17
CLM	+ 0760. . .000	2	Clear Magnitude	19
CLS	+ 0502	2	Clear and Subtract	18
COM	+ 0760. . .006	2	Complement Magnitude	20
CPY	+ 0700	– III	Copy and Skip	27
DCT	+ 0760. . .012	2	Divide Check Test	25
DVH	+ 0220	20	Divide or Halt	18
DVP	+ 0221	20	Divide or Proceed	18
ETM	+ 0760. . .007	2	Enter Trapping Mode	23
ETT	— 0760. . .011	2	End of Tape Test	27
FAD	+ 0300	7 II	Floating Add	21
FDH	+ 0240	18 IV	Floating Divide or Halt	22
FDP	+ 0241	18 IV	Floating Divide or Proceed	23
FMP	+ 0260	17	Floating Multiply	22
FSB	+ 0302	7 II	Floating Subtract	22
HPR	+ 0420	2	Halt and Proceed	23
HTR	+ 0000	2	Halt and Transfer	24
LBT	+ 0760. . .001	2	Low Order Bit Test	25
LDA	+ 0460	2	Locate Drum Address	27
LDQ	+ 0560	2	Load MQ	19
LGL	— 0763	2 I	Logical Left	21
LLS	+ 0763	2 I	Long Left Shift	20
LRS	+ 0765	2 I	Long Right Shift	21
LTM	— 0760. . .007	2	Leave Trapping Mode	23
LXA	+ 0534	2	Load Index from Address*	26
LXD	— 0534	2	Load Index from Decrement*	26
MPR	— 0200	20	Multiply and Round	18
MPY	+ 0200	20	Multiply	18
MSE	— 0760	2	Minus Sense	27
NOP	+ 0761	2	No Operation	23
ORA	— 0501	2	OR to Accumulator	20
ORS	— 0602	2	OR to Storage	20
PAX	+ 0734	2	Place Address in Index*	26
PBT	— 0760. . .001	2	P Bit Test	25
PDX	— 0734	2	Place Decrement in Index*	26
PSE	+ 0760	2	Plus Sense	27
PXD	— 0754	2	Place Index in Decrement*	26
RDS	+ 0762	2 III	Read Select	26
REW	+ 0772	40 ms III	Rewind	27
RND	+ 0760. . .010	2	Round	18
RQL	— 0773	2 I	Rotate MQ Left	21
RTT	— 0760. . .012	2	Redundancy Tape Test	25
SBM	— 0400	2	Subtract Magnitude	18
SLQ	— 0620	2	Store Left-Half MQ	19
SLW	+ 0602	2	Store Logical Word	19
SSM	— 0760. . .003	2	Set Sign Minus	19

Alpha Code	Octal Code	Cycles	Operation	Page
SSP	+ 0760. . .003	2	Set Sign Plus	19
STA	+ 0621	2	Store Address	19
STD	+ 0622	2	Store Decrement	19
STO	+ 0601	2	Store	19
STP	+ 0630	2	Store Prefix	19
STQ	— 0600	2	Store MQ	19
SUB	+ 0402	2	Subtract	18
SXD	— 0634	2	Store Index in Decrement*	26
TIX	+ 2000	2	Transfer on Index**	25
TLQ	+ 0040	2	Transfer on Low MQ	24
TMI	— 0120	2	Transfer on Minus	24
TNO	— 0140	2	Transfer on No Overflow	24
TNX	— 2000	2	Transfer on No Index**	25
TNZ	— 0100	2	Transfer on No Zero	24
TOV	+ 0140	2	Transfer on Overflow	24
TPL	+ 0120	2	Transfer on Plus	24
TQO	+ 0161	2	Transfer on MQ Overflow	24
TQP	+ 0162	2	Transfer on MQ Plus	24
TRA	+ 0020	2	Transfer	24
TSX	+ 0074	2	Transfer and Set Index*	24
TTR	+ 0021	2	Trap Transfer	25
TXH	+ 3000	2	Transfer on Index High**	25
TXI	+ 1000	2	Transfer with Index Incremented**	25
TXL	— 3000	2	Transfer on Index Low or Equal**	25
TZE	+ 0100	2	Transfer on Zero	24
UFA	— 0300	6 II	Unnormalized Floating Add	22
UFM	— 0260	17	Unnormalized Floating Multiply	22
UFS	— 0302	6 II	Unnormalized Floating Subtract	22
WEF	+ 0770	2 III	Write End of File	26
WRS	+ 0766	2 III	Write Select	26

* Not indexable.
** Not indexable but contains a decrement part.

INDEX

IBM

Form 24-6661-2 (2-57, 10M—ME)